N° d'ordre : 1088

THÈSE

DE DOCTORAT D'ÉTAT ÈS-SCIENCES PHYSIQUES

PRÉSENTÉE À L'UNIVERSITÉ DE PARIS-XI

PAR

Hervé DELOUIS

Centre Inter-Régional de Calcul Electronique

C.N.R.S.

POUR OBTENIR

LE GRADE DE DOCTEUR ÈS-SCIENCES

Sujet : MISE AU POINT D'UNE CHAÎNE DE PROGRAMMES PERMETTANT LE CALCUL ET L'EXPLOITATION AUTOMATIQUE DES SPECTRES OBTENUS PAR TRANSFORMATION DE FOURIER

Soutenue le 20 mars 1973 devant la commission d'examen

M. P. JACQUINOT — PRÉSIDENT

MM. M. FRANÇON
J. ARSAC
Mme J. CONNES
— EXAMINATEURS

Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41544713_0019

N° d'ordre : 1088

THÈSE

DE DOCTORAT D'ÉTAT ÈS-SCIENCES PHYSIQUES

PRÉSENTÉE À L'UNIVERSITÉ DE PARIS-VI

PAR

Hervé DELOUIS

Centre Inter-Régional de Calcul Electronique

C.N.R.S

POUR OBTENIR

LE GRADE DE DOCTEUR ÈS-SCIENCES

Sujet : MISE AU POINT D'UNE CHAÎNE DE PROGRAMMES PERMETTANT LE CALCUL ET L'EXPLOITATION AUTOMATIQUE DES SPECTRES OBTENUS PAR TRANSFORMATION DE FOURIER

Soutenue le 20 mars 1973 devant la commission d'examen

M. P. JACQUINOT — PRÉSIDENT

MM. M. FRANÇON
J. ARSAC
Mme J. CONNES — EXAMINATEURS

UNIV.
LEIDEN
BIBL.

TABLE DES MATIERES

INTRODUCTION 1

I. CALCUL NUMERIQUE DE LA TRANSFORMEE DE FOURIER (T) D'UNE FONCTION REELLE

I.1. GENERALITES 8

I.2. TRANSFORMEE DE FOURIER D'UNE FONCTION LIMITEE 9

- Définitions et notations
- Rappel du théorème sur la T.F. du produit de deux fonctions
- Fonction d'appareil

I.3. TRANSFORMEE DE FOURIER D'UNE FONCTION ECHANTILLONNEE 13

- Théorème d'échantillonnage
- T.F. d'une distribution de Dirac de période h
- Choix du pas d'échantillonnage

I.4. TRANSFORMEE DE FOURIER D'UNE FONCTION LIMITEE ET ECHANTILLONNEE 15

I.5. TRANSFORMEE DE FOURIER DISCRETE NORMALISEE (DFTN) 17

I.5.1. Transformée de Fourier discrète DFT 17
I.5.2. Transformée de Fourier discrète normalisée DFTN 20

α. Importance de l'emplacement des échantillons
β. Normalisation du pas d'échantillonnage (homothétie)

II. UTILISATION DE L'ALGORITHME F.F.T. POUR LE CALCUL DE LA DFTN D'UNE FONCTION REELLE IMPAIRE 23

II.1. ETUDE THEORIQUE

II.1.1. Définition de la F.F.T. par la technique dite de "decimation in time"
II.1.2. Adaptation de l'algorithme "F.F.T." au cas de la transformée d'une fonction réelle 25
- schéma II
II.1.3. Adaptation de l'algorithme "F.F.T." au cas de la transformée d'une fonction réelle impaire (ou paire) 28

- DFTN d'une fonction hermitique (schéma III) 31
- DFTN d'une fonction réelle impaire (schéma IV) 32

II.2. PROGRAMMATION DE LA FFT

II.2.1. Transformation d'une fonction complexe en mémoire centrale. Programme "FFT32T". 36

α. Première partie, le tri initial des données 37

β. Deuxième partie, le calcul des intercombinaisons 40

II.2.2. Transformation d'une fonction réelle en mémoire centrale. Programme "FFT64K" 42

α. Première partie, constitution de $\{En\}_N$ à partir de $\{In\}_{N'}$.

β. Deuxième partie, le calcul de la $DFTN^{+}_{(N)}$.

γ. Troisième partie, le calcul de la fonction hermitique.

II.2.3. Transformation d'une fonction réelle impaire en mémoire centrale. Programme "FFT128" 45

α. Constitution de la fonction hermitique H.

β. Calcul de la DFTN de la fonction hermitique H.

γ. Le calcul final des N valeurs B_k.

II.2.4. Transformation d'une fonction réelle impaire nécessitant l'utilisation d'un support auxiliaire de mémoire sur disque magnétique. Programme "TFID4" 50

α. Principe de base. 51

β. Acquisition des données et constitution de la fonction hermitique H. 54

γ. Constitution de $\{F_n\}_{N/2}$. 56

δ. Sélection et tri des n_b séries dont il faut calculer la DFTN. 56

ε. Calcul du spectre $\{B_k\}_{N'}$ à partir de $\{E_k\}_{N/2}$. 60

III. INTERPOLATION DANS LE SPECTRE

III.1. ETUDE THEORIQUE

III.1.1. Principe de l'interpolation sans apodisation. 61

III.1.2. Principe de l'interpolation avec apodisation. 63

III.2. REALISATION PRATIQUE DE L'INTERPOLATION 65

III.2.1. Précaution concernant le calcul des valeurs primaires $B(\nu)$. 66

III.2.2. Pondération de la fonction d'interpolation $f(\nu)$ 67

III.2.3. L'algorithme du programme d'interpolation. 67

α. Choix de la dimension du tableau B des valeurs primaires. 71

β. Calcul de la fonction d'interpolation $f(\nu) \rightarrow \{A_j\}$ 72

- Cas particulier de l'interpolation sans apodisation.
- Justification du choix de la valeur $N_p = 201$. 74
- Différents types d'apodisation.
- Programme d'interpolation et de calcul de $f(\nu)$ 76

III.3. RESULTATS NUMERIQUES 78

IV. PROBLEMES SPECIFIQUES DE LA SPECTROSCOPIE PAR TRANSFORMATION DE FOURIER

IV.1. ACQUISITION DES DONNEES - ENREGISTREMENT ET LECTURE 82

IV.1.1. Généralités

IV.1.2. Utilisation des bandes magnétiques 7 pistes (556 bpi) 84

α. Schéma de codage.
β. Lecture et décodage des valeurs.
γ. Inconvénients.

IV.1.3. Utilisation des bandes magnétiques 9 pistes (800 bpi) 87

α. Schéma de codage.
β. Lecture et décodage des valeurs
- Sous-programme "HERLEC" (liste)
γ. Avantages de l'enregistrement sur 9 pistes. 94

FIGURES relatives à ces 4 chapitres. 96

CONCLUSION 106

REFERENCES 109

LISTE DES PRINCIPAUX PROGRAMMES 111

FFT32T - V3 111
FFT64K - V3 113
FFT128 - V3 116
TFID4 120
MPLØT 124
BSS1Ɔ 126
SP5 (TEST8) 128
T102 131

IV.2. BROCHURE CONCERNANT LE PRINCIPE ET LE MODE D'EMPLOI DU PROGRAMME "DERPTE"

qui recherche automatiquement les raies d'un spectre et en calcule la position précise ((*)) (feuilles orange).

IV.3. BROCHURE "MANUEL DU PARFAIT PETIT FOURIERISTE"

concernant le mode d'emploi des autres programmes d'exploitation ((*)) (feuilles bleues).

((*)) Pour ces deux brochures, se reporter à leur propre table des matières.

INTRODUCTION

En spectroscopie par transformation de Fourier, on enregistre toujours un interférogramme qui est la transformée de Fourier en cosinus ou en sinus du spectre étudié suivant que la technique de modulation interne [1] est appliquée ou non.

Le traitement numérique sur ordinateur de l'interférogramme peut être fait par une chaîne de programmes dans laquelle on peut distinguer cinq parties distinctes.

1. L'acquisition des données.

Cette première partie doit permettre la lecture, le décodage, la vérification et le stockage de l'échantillonnage à pas constant de l'interférogramme qui en général a été enregistré sur ruban perforé ou bande magnétique.

2. La correction ou l'adaptation éventuelle de l'interférogramme.

Le calcul du spectre par une transformée de Fourier en cosinus ou en sinus suppose que le premier échantillon se trouve sur l'axe ou le point de symétrie de l'interférogramme (qui correspond à la différence de marche zéro). Si ce n'est pas le cas, il faut d'abord évaluer le décalage des échantillons, puis calculer par application de la théorie d'interpolation ceux qui partent effectivement du point de symétrie.

Si à des fins de test, on ne veut calculer qu'une partie du spectre, c'est à ce niveau qu'on peut réaliser une opération de filtrage en calculant les valeurs plus espacées de l'interférogramme qui correspondrait à la tranche spectrale étroite désirée.

3. Le calcul de la transformée de Fourier en cosinus ou en sinus de l'interférogramme.

Il permet le calcul du nombre minimum de "valeurs primaires" du spectre qui, d'après le théorème d'échantillonnage, contiennent toute l'information.

4. Interpolation - Apodisation du spectre.

L'exploitation et en particulier le tracé du spectre ne peut se faire directement à partir des valeurs primaires qui sont trop espacées pour l'étude visuelle et numérique de la forme des raies. L'interpolation permet donc d'obtenir un échantillonnage plus serré (cinq fois en général). Nous les appellerons valeurs secondaires.

L'apodisation de la fonction d'appareil dans le spectre peut être faite en pondérant l'interférogramme avant le calcul de la T.F. mais aussi au moment de l'interpolation des points primaires.

5. Exploitation du spectre.

Il s'agit des programmes permettant par exemple :

- de faire des moyennes de spectres analogues pour augmenter le rapport signal sur bruit.

- de faire le tracé plus ou moins élaboré du spectre.

- de faire une recherche et une étude automatique de la position et de forme des raies.

Au cours des 15 dernières années, la chaîne de programmes ainsi définie a dû s'adapter parallèlement à l'évolution technique des interféromètres qui s'est faite en quatre étapes principales que nous allons rappeler brièvement. [2]

1ère étape (1959-1960)

L'interféromètre de Michelson construit par J. CONNES au laboratoire

Aimé Cotton permettait une différence de marche maximum Δ_M de 1 cm. La limite de résolution $\delta\sigma = \frac{1}{\Delta_M}$ était donc de l'ordre du cm^{-1}. Le nombre M d'échantillons de l'interférogramme était de l'ordre de 10.000 et à l'aide d'une transformée de Fourier en cosinus classique, le calcul de NS = 1500 valeurs du spectre prenait environ 5 heures sur IBM 704. L'ensemble des problèmes théoriques posés par le calcul numérique du spectre à partir de l'interférogramme échantillonné a été développé par J. CONNES dans sa thèse [3].

Au point de vue programmation, notons que :

- les M échantillons étaient relevés sur l'interférogramme enregistré en continu.

- Le calcul de la T.F. en cosinus était fait de façon classique (durée proportionnelle à M x NS où NS = nombre de points du spectre calculé.

- les premiers essais de filtrage numérique pour isoler la région spectrale intéressante et réduire le nombre de points utilisés pour faire la T.F. ont été faits à cette époque [4].

2ème étape (1964-1966)

L'interféromètre construit par P. CONNES avec la collaboration du Jet Propulsion Laboratory [1,5,6] permettait une différence de marche Δ_M de 10 cm ce qui correspondait à une limite de résolution $\delta\sigma$ de 0,1 cm^{-1}. Avec cet interféromètre, l'interférogramme impair était échantillonné avec M = 64.000 valeurs enregistrées sur ruban perforé. Le nombre de points primaires à calculer dans le spectre était aussi de 64.000 valeurs.

Vis à vis des 5 parties que nous avons distinguées dans la chaîne des programmes d'exploitation, nous avons alors développé.

Pour la 1ère partie. Un programme de décodage et de vérification des échantillons lus sur ruban perforé. Une fois décodées ces valeurs étaient stockées sur bande magnétique.

Pour la 2ème partie. Deux programmes différents pour la mesure du déphasage (ou décalage) de l'interférogramme [7].

Un programme de correction du déphasage mesuré.

Ces trois programmes nécessitaient la connaissance d'environ 300 valeurs de l'interférogramme avant la différence de marche zéro.

Un programme de test, qui à partir du ruban perforé et après filtrage numérique, permettait le calcul rapide d'une ou plusieurs tranches spectrales [7].

Pour la 3ème partie. Un programme "JPAUL" de calcul de T.F. d'une fonction complexe utilisant l'algorithme F.F.T. de Cooley et Tuckey. Ce programme entièrement en mémoire centrale ne permettait pas de calculer la T.F. d'un interférogramme de plus de 4096 valeurs sur l'ordinateur 7040 IBM de 32 K ((*)) mots dont nous disposions.

Un programme de calcul de T.F. utilisant aussi l'algorithme F.F.T. mais nécessitant quatre bandes magnétiques pour pallier au manque de place en mémoire centrale. Avec ce programme le calcul de 64 K valeurs primaires prenait 105 minutes sur le 7040 IBM.

Pour la 4ème partie. Un programme d'interpolation qui permettait aussi l'apodisation du spectre.

Pour la 5ème partie. Un programme de tracé "MPLØT" sur traceur Benson.

Un programme de moyennes de spectres de même nature, mais de résolution éventuellement différentes [7].

Un programme de mesure précise de la position de raies relevées grossièrement sur un spectre déjà tracé.

3ème étape (1965-1968)

L'interféromètre construit par Pinard [8], [9] permettait d'atteindre une différence de marche Δ_M de 1,60 m ($\delta\sigma = 6.10^{-3}$ cm^{-1}), mais le domaine spectral étudié ne dépassait pas 400 cm^{-1}. Cette caractéristique permettait l'enregistrement d'interférogrammes avec un pas d'échantillonnage très grand ce qui limitait le nombre M d'échantillons à 64.000.

((*)) K = 2^{10} = 1024

Les calculs relatifs à ce tvpe d'interféromètre ont été faits avec la chaîne de programmes qui avait été mise au point lors de la 2ème étape.

4ème étape (A partir de 1969)

Il s'agit des interféromètres construits pour l'étude de larges domaines spectraux (16.000 cm^{-1}) à une résolution très grande (différence de marche de l'ordre de quelques mètres). L'interférogramme est une fonction impaire échantillonnée à partir du centre de symétrie.

Le nombre d'échantillons est de l'ordre de 2.10^6 et il en est de même du nombre d'éléments spectraux dans le spectre. De plus le nombre d'interférogrammes à traiter est assez élevé puisque quatre interféromètres de ce type fonctionnent déjà au laboratoire Aimé Cotton et que d'autres sont en construction dans d'autres laboratoires.

Pour pouvoir exploiter une telle quantité d'informations, voyons dans l'ordre où elles sont exposées par la suite, les principales améliorations qui ont dû être apportées à la chaîne de programmes d'exploitation déjà décrite.

Un premier objectif a été, d'une part de rendre le temps de calcul de la T.F. pratiquement négligeable vis à vis de la durée totale des calculs nécessaires à l'exploitation proprement dite du spectre, et d'autre part de permettre ce calcul indépendamment de la taille de la mémoire centrale de l'ordinateur dont on dispose. Dans ce but, nous avons commencé par écrire des sous-programmes (F.F.T.) nettement plus rapides pour calculer entièrement en mémoire centrale la T.F. d'une fonction complexe ("F.F.T.32T"), d'une fonction réelle ("F.F.T.64K") et d'une fonction réelle impaire ("F.F.T.128"). Enfin le programme "TFID4" a été écrit pour permettre le calcul de la T.F. en sinus d'une fonction réelle impaire de plusieurs millions de points et ce, quelle que soit la taille de la mémoire centrale. Il utilise des disques magnétiques comme support auxiliaire de mémoire. Avant de décrire ces algorithmes, nous commencerons par rappeler la théorie du calcul de la transformée de Fourier discrète (D.F.T.) d'une fonction réelle quelconque puis impaire, en la rattachant au théorème d'échantillonnage et à la valeur au sens physique du spectre ainsi calculé. L'ensemble des quatre programmes de T.F.

est programmé en Fortran et constitue un ensemble cohérent en ce sens que les algorithmes définis à propos des trois premiers ont servi à l'élaboration du quatrième. Le calcul par "TFID4" de la D.F.T. de 2.10^6 échantillons dure environ 8 minutes sur le 370/165 du C.I.R.C.E. On aura une idée des progrès réalisés dans les techniques de programmation de la D.F.T. en sachant qu'avec une méthode classique de T.F. le même calcul sur le même ordinateur durerait environ 4 années.

Un deuxième objectif a été de réduire la durée de l'interpolation des points primaires du spectre calculés par TFID4 ou FFT128, et de profiter de cette interpolation pour introduire des apodisations diverses pour la fonction d'appareil. Au niveau des archives de spectres, on peut ainsi se contenter de stocker sur bande magnétique les valeurs primaires non apodisées. Le gain de temps a été obtenu d'une part par l'amélioration des techniques de programmation (en Fortran) et d'autre part par une minimisation de la durée du calcul de la fonction d'interpolation et des valeurs secondaires tout en conservant une précision suffisante pour ces derniers. Sur le 370/165 IBM l'interpolation (10^7 points secondaires) de 2.10^6 valeurs primaires se fait ainsi en moins de 7 minutes.

Un troisième objectif a été d'améliorer toute la partie acquisition de données sur trois plans : la fiabilité, la capacité de stockage, et la rapidité tant au stade de l'enregistrement que de la lecture et du décodage des valeurs de l'interférogramme. Ces améliorations ont été obtenues d'une part grâce à l'évolution des techniques d'enregistrement (remplacement du ruban perforé par des bandes magnétiques 7 puis 9 pistes par exemple) et d'autre part par un codage qui soit mieux adapté aux ordinateurs utilisés (pas de gaspillage de place) et qui permette une vérification de la validité des valeurs enregistrées (numérotation des mesures depuis l'utilisation des bandes 9 pistes).

Un quatrième objectif plus général a été d'automatiser et de faciliter l'utilisation de l'ensemble de la chaîne de programmes d'exploitation qui sont utilisés par de nombreux chercheurs non spécialisés en informatique. Dans ce but, des efforts pour simplifier et standardiser les cartes données ont été faits. L'ensemble des programmes a été mis en bibliothèque sur disque magnétique au C.I.R.C.E. et des procédures d'enchaînement de programmes ont

été mises au point. Deux brochures assez complètes ont été faites pour définir les objectifs et les possibilités de ces programmes et permettre leur utilisation.

La première brochure est relative au programme "DERPTE" qui détecte automatiquement les raies d'un spectre d'émission ou d'absorption et détermine leur position très précise. Il peut aussi en option calculer la largeur équivalente de chaque raie. Les algorithmes employés y sont largement exposés de façon à ce que les utilisateurs puissent éventuellement l'adapter à des cas particuliers.

La deuxième brochure ("Le parfait petit fouriériste") est relative à l'ensemble des autres programmes d'exploitation. Son but est surtout utilitaire et il faut se reporter pour certains (non exposés ici) à d'autres publications pour ce qui est de la théorie complète. Nous y trouvons successivement le mode d'emploi des programmes :

"DUMPBDE" qui donne une image non décodée de l'information stockée sur une bande (en cas d'erreurs).

"TEST8" qui a des fins de tests permet le calcul très rapide de tranches spectrales étroites après filtrage numérique.

"BSS10" qui permet le calcul du spectre à basse résolution.

"TFID3" déjà vu plus haut.

"MPLØT" qui est utilisé pour tracer le spectre.

D'autres programmes "utilitaires" sont décrits et pour plus de détails il suffit de se reporter à la table des matières propre à chaque brochure((*)). La liste de la plupart des programmes est fournie.

((*)) Un détail pratique : Ces deux brochures ont leur propre pagination mais se distinguent du reste de l'exposé par la couleur du papier.

I.1. - GENERALITES SUR LE CALCUL NUMERIQUE DE LA TRANSFORMEE DE FOURIER (T) D'UNE FONCTION REELLE.

- Les interférogrammes que nous devons transformer, sont, en l'absence de bruit, toujours des fonctions réelles impaires à spectre limité.

- Au lieu d'étudier directement la transformation d'une telle fonction, nous la traiterons plutôt comme un cas particulier de la transformation de Fourier d'une fonction réelle sans caractère de symétrie de façon à bien faire ressortir les avantages du premier type de fonction.

- Pour ne pas trop s'éloigner du problème physique qui nous intéresse, nous imposons une seule contrainte à la fonction réelle I(t) dont il faut calculer numériquement la T.F.

I(t) a un spectre limité ; autrement dit, sa T.F. est limitée à l'intérieur d'un domaine $\pm\Delta\nu$ de fréquence ν. Une conséquence de cette dernière contrainte est que les variations de I(t) autour de sa valeur moyenne deviennent d'autant plus faibles en valeur absolue qu'on s'éloigne de l'origine t=o et que $\Delta\nu$ est plus large.

Le calcul numérique de la T.F. de I(t) se fera forcément à partir d'un échantillonnage fini de cette fonction. Nous commencerons donc par étudier l'influence sur la T.F. de la limitation et de l'échantillonnage d'une fonction.

Cet échantillonnage fini sera en fait traité par des programmes dits de T.F. En fait, ces programmes qui utilisent depuis quelques années l'algorithme "F.F.T." de Cooley Tuckey reposent sur une définition normalisée de la T.F. discrète que nous symboliserons par "D.F.T.N." (discrete fourier transform normalised). Nous définirons donc cette "D.F.T.N." et nous exposerons brièvement le principe de la "F.F.T.".

Nous verrons alors comment se situe le cas particulier d'une fonction impaire par rapport au cas général d'une fonction quelconque. Les avantages mis en évidence sont en fait équivalents à ceux qu'on peut attendre d'une détection synchrone par rapport à une détection classique linéaire ou quadratique.

I.2. - TRANSFORMEE DE FOURIER D'UNE FONCTION LIMITEE

Définitions et Notations : La transformée de Fourier B(ν) d'une fonction réelle ou complexe I(t) est définie par :

$$(1) \qquad B(\nu)=T^{+}\left[I(t)\right] = \int_{-\infty}^{+\infty} I(t)e^{2i\pi\nu t}\,dt \quad (\text{avec } t \text{ et } \nu \in R)$$

cette expression a toujours un sens si $I(t) \in L$, c'est-à-dire si :

$$\int_{-\infty}^{+\infty} |I(t)|dt = ||I|| \text{ existe} \rightarrow B(\nu) \text{ uniformément bornée.}$$

Du fait de la réciprocité de cette transformation, nous aurons :

$$(2) \qquad I(t)=T^{-}\left[B(\nu)\right] = \int_{-\infty}^{+\infty} B(\nu)e^{-2i\pi\nu t}\,d\nu$$

Il est souvent utile de considérer une transformée de Fourier (ou T.F.), comme la somme d'une transformée en cosinus T_c et d'une transformée en sinus T_s.

$$(3) \qquad \begin{cases} T^{+} = T_c + i\,T_s \\ T^{-} = T_c - i\,T_s \end{cases}$$

$$\text{avec} \quad T_c\left[I(t)\right] = \int_{-\infty}^{+\infty} I(t)\cos 2\pi\nu t\,dt$$

$$T_s\left[I(t)\right] = \int_{-\infty}^{+\infty} I(t)\sin 2\pi\nu t\,dt$$

Soit $I_1(t)$ une fonction réelle définie de $-\infty$ à $+\infty$ (fig. 1,a). Sa T.F. $B_1(\nu)$ est hermitique et comme on l'a vu dans les généralités qui précèdent, son domaine est limité aux fréquences $\pm\Delta\nu$.

$$(4) \qquad B_1(\nu)= T^{+}\left[I_1(t)\right]$$

$$\text{et} \quad I_1(t) \text{ réel} \longleftrightarrow B_1^{*}(\nu)= B_1(-\nu)$$

$$\text{D'où :} \quad B_1(\nu)=B_{1p}(\nu)+i\,B_{1i}(\nu)$$

$$\text{avec} \quad \begin{cases} B_{1p}(\nu) = T_c[I_1(t)] & \text{réelle et paire} \\ B_{1i}(\nu) = T_s[I_1(t)] & \text{réelle et impaire} \end{cases} \tag{5}$$

Le domaine spectral $\pm\Delta\nu$ ayant toujours une certaine largeur, la valeur moyenne de $I_1(t)$ est négligeable et en valeur absolue cette fonction décroît plus ou moins rapidement dès que $|t|$ augmente. D'un point de vue pratique, la connaissance de $I_1(t)$ sera toujours limitée à un domaine fini. Ce domaine sera en général centré sur l'origine pour des raisons évidentes de rapport signal sur bruit et pour des raisons physiques (dans le cas d'un interférogramme, t représente le déplacement d'un miroir de part et d'autre de la position correspondant à une différence de marche nulle). Cette limitation équivaut à multiplier $I_1(t)$ par une "fonction rectangle" $D_{2T}(t)$ de largeur $2T$ centrée sur l'origine t=o.

$$\text{avec} \quad \begin{cases} D_{2T} = 1 & \text{si } -T \leqslant t \leqslant +T. \\ D_{2T} = 0 & \text{partout ailleurs.} \end{cases}$$

On définit ainsi une nouvelle fonction $I_2(t)$ (fig. 1,b)

$$I_2(t) = I_1(t) . D_{2T}(t) \tag{6}$$

$$B_2(\nu) = T^+[I_2(t)] = \int_{-\infty}^{+\infty} I_2(t) e^{2i\pi\nu t} dt = \int_{-T}^{+T} I_2(t) e^{2i\pi\nu t} dt \tag{7}$$

Son spectre $B_2(\nu)$ est calculé à partir d'une intégrale sur un domaine fini $[-T, +T]$. Voyons dans quelle mesure il diffère de $B_1(\nu)$?

RAPPEL : Théorème concernant la T.F. du produit de deux fonctions f(t) et g(t).

Soit f(t) et G(ν) deux fonctions $\in$ L (I.2.(2)).

$$\text{(8) Si} \quad \begin{cases} F(\nu) = T^+[f(t)] \\ g(t) = T^-[G(\nu)] \end{cases} \quad \text{alors :} \quad \begin{cases} T^-[F.G] = f * g = h(t) \\ T^+[f.g] = F * G \end{cases}$$

$h=f * g$ est le produit de convolution des deux fonctions f et g défini par :

$$h(t_o) = \int_{-\infty}^{+\infty} f(t) g(t_o - t)\, dt$$

Appliquons cette règle au produit des deux fonctions I_1 et D_{2T} de l'expression (6).

(9) $$B_2(\nu) = B_1(\nu) * f_{2T}(\nu)$$

(10) avec $$f_{2T}(\nu) = T^+ \left[D_{2T}(t)\right] = T_c \left[D_{2T}(t)\right] = 2T \frac{\sin \pi\nu 2T}{\pi\nu 2T}$$

L'expression (9) peut aussi s'écrire :

(11) $$B_2(\nu_o) = \int_{-\infty}^{+\infty} B_1(\nu)\, f_{2T}(\nu_o - \nu)\, d\nu$$

Sous cette forme, on voit que la valeur $B_2(\nu_o)$ est égale à la surface découpée dans $B_1(\nu)$ par la fonction $f_{2T}(\nu)$ qu'on appelle souvent fonction d'appareil.

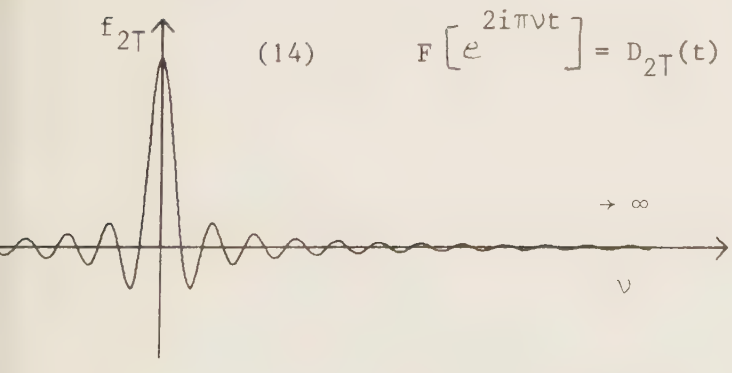
Si le produit de convolution décrit par (9) est assimilé à la notion générale de filtre linéaire F, on dira que $f_{2T}(\nu)$ est la réponse percussionnelle de ce filtre F et que $D_{2T}(t)$ en est le gain, ou la valeur propre vis à vis des fonctions $\{ e^{2i\pi\nu t} \}$ qui en sont les fonctions propres.

On aura alors :

(12) $$B_2(\nu) = F\left[B_1(\nu)\right] = B_1(\nu) * f_{2T}(\nu)$$

(13) $$f_{2T}(\nu) = F\left[\delta(\nu)\right] \quad \text{avec :} \quad \delta(\nu) = \begin{cases} \text{impulsion} \\ \text{fonction de Dirac} \end{cases}$$

(14) $$F\left[e^{2i\pi\nu t}\right] = D_{2T}(t)\, e^{2i\pi\nu t} \quad \forall t$$



- $f_{2T}(\nu)$ est constituée d'un pic central dont la largeur à mi-hauteur est $\delta\nu \simeq 1/2\ T$ et ce pic est encadré par des oscillations qui s'étendent jusqu'à l'infini en décroissant lentement. D'après ces caractéristiques il est facile de voir quels seront les deux principaux effets de ce filtrage linéaire.

a) Le spectre ne sera plus nul en dehors du domaine $\pm\Delta\nu$.

b) La résolution dans le spectre n'est plus infinie.

La perte de résolution peut être mise en évidence de trois façons équivalentes :

- Par l'équation (11) et son interprétation.
- Par l'équation (13) qui montre qu'une impulsion $\delta(t)$ de largeur infiniment petite est remplacée par la réponse percussionnelle de largeur $\delta\nu$; $\delta\nu$ étant la limite de résolution.
- Enfin par l'équation (14) qui montre que les composantes du spectre extérieures à $D_{2T}(t)$ sont éliminées.

En conséquence, nous dirons que deux impulsions $\delta(\nu-\nu_1)$ et $\delta(\nu-\nu_2)$ présentes dans $B_1(\nu)$ ne pourront être résolues dans $B_2(\nu)$ que si $|\nu_2-\nu_1| > \delta\nu$.

Si c'est la largeur $\delta\nu$ du pic central qui est responsable de la perte de résolution, les oscillations latérales, elles, vont déformer le spectre et étendre son domaine au delà des fréquences $\pm\Delta\nu$. Ces derniers effets peuvent être diminués car il est possible d'obtenir une fonction d'appareil $f(\nu)$ dont les oscillations latérales soient beaucoup plus faibles en multipliant le rectangle D_{2T} par une fonction d'apodisation convenable.

Par exemple :

$$D'_{2T}(t) = D_{2T}(t)\left[1 - \frac{t2}{T2}\right]^2 \qquad (15)$$

La diminution des "pieds" de la fonction d'appareil s'accompagne d'une légère augmentation de la largeur du pic central et il faut donc, suivant les cas, trouver un compromis optimum entre ces deux phénomènes. [3]

REMARQUE : Sur la figure (1,b) on peut voir que les perturbations introduites par les oscillations de $f_{2T}(\nu)$ sont très faibles ce qui est toujours le cas lorsque le spectre s'étend sur un domaine $\Delta\nu >> \delta\nu$.

A titre indicatif, le rapport $\Delta\nu/\delta\nu$ est égal à 16 dans le cas de la figure (1,b).

N.B. : Toutes les courbes des figures 1 et 2 ont été normalisées avant d'être tracées.

I.3. - TRANSFORMEE DE FOURIER D'UNE FONCTION ECHANTILLONNEE

Par hypothèse, la fonction $I_1(t)$ a un spectre limité au domaine $\pm\Delta\nu$. Nous pouvons donc lui appliquer le théorème d'échantillonnage [3, 10 à 15] que nous allons rappeler brièvement.

- Si une fonction I(t) est à spectre limité de support $[-\nu_1, +\nu_1]$, toutes les informations concernant cette fonction sont contenues dans une infinité dénombrable d'échantillons de I(t) aux instants :

$$t_n = \frac{n}{2\nu_1} \quad (\rightarrow \text{pas d'échantillonnage } h = \frac{1}{2\nu_1})$$

appelés instants d'échantillonnage.

On montre que la position des échantillons $\{t_n\}$, pourvu qu'ils restent périodiques, est sans importance.

Les autres valeurs de la fonction peuvent être obtenues par la convolution :

$$(16) \qquad I(t) = \sum_{n=-\infty}^{+\infty} I(t_n)\ \mathrm{sinc}\left[2\nu_1 (t-t_n)\right]$$

où $\mathrm{sinc}(t) = \dfrac{\sin\pi t}{\pi t}$

L'expression de ce théorème, est limitée au cas particulier où le spectre est nul en dehors du domaine $[-\nu_1, +\nu_1]$. Dans le cas ou le spectre est à bande limitée $[-\nu_2, -\nu_1]$ et $[\nu_1, \nu_2]$ l'expression (16) devient : [16, 7]

$$(17) \qquad I(t) = \sum_{n=-\infty}^{+\infty} I(t_n)\ \mathrm{sinc}\left[\Delta\nu(t-t_n)\right] \cos 2\pi\nu_m\ (t-t_n)$$

avec $\Delta\nu = \nu_2 - \nu_1$ et $\nu_m = \dfrac{\nu_1 + \nu_2}{2}$

Il est donc tout a fait justifié de vouloir échantillonner notre fonction $I_1(t)$. Mais afin de bien montrer l'importance du choix correct du pas h, nous allons montrer quelles sont les conséquences pour le spectre d'un échantillonnage quelconque.

Soit $I_3(t)$ l'ensemble de valeurs de $I_1(t)$ prises pour des valeurs de t égales à : $t_n = nh$.

$$(18) \qquad I_3(t) = \sum_{n=-\infty}^{+\infty} I_1(t)\ \delta(t-t_n) \qquad (\delta(t) = \text{fonction de Dirac})$$

si on pose $R_h(t) = \sum_{n=-\infty}^{+\infty} \delta(t-nh) = \sum_{n=-\infty}^{+\infty} \delta(t-t_n)$

$$(19) \qquad I_3(t) = I_1(t)\ R_h(t)$$

$R_h(t)$ est une distribution de Dirac de période h. D'après (8), nous pouvons transformer (19) et obtenir :

$$B_3(\nu) = T^+\left[I_1(t)\right] * T^+\left[R_h(t)\right]$$

La T.F. d'une distribution de Dirac de période h est une autre distribution de période 1/h.

$$(20) \qquad T^+\left[R_h(t)\right] = \frac{1}{h}\ R_{1/h}(\nu) = \frac{1}{h}\ \sum_{k=-\infty}^{+\infty} \delta(\nu - \frac{k}{h})$$

On obtient alors au coefficient 1/h près :

$$(21) \qquad \boxed{B_3(\nu) = B_1(\nu) * R_{1/h}(\nu)}$$

$B_3(\nu)$ devient alors une fonction périodique de période 1/h. (fig. 1,c), et si on a bien choisi la valeur du pas d'échantillonnage h, on retrouve dans chacune de ses périodes le spectre $B_1(\nu)$. C'est le cas de la figure 1 où h vaut 1/64 et 1/2h vaut 32.

CHOIX DU PAS D'ECHANTILLONNAGE h.

Soit $\Delta\nu = \left[\nu_1, \nu_2\right] = \left[-\nu_2, -\nu_1\right]$ la largeur du domaine spectral de $B_1(\nu)$. Chaque période de $B_3(\nu)$ est constituée d'une demi-période à gauche et d'une demi-période à droite. Supposons que $\left[\nu_1, \nu_2\right]$ soit entièrement compris dans une de ces demi-périodes, celle de gauche par exemple comme dans le cas de la figure (1,c). Du fait de la symétrie par rapport à l'origine de $B_1(\nu)$ (relation (5)), le domaine $\left[-\nu_2, -\nu_1\right]$ est lui, à l'intérieur d'une des demi-périodes de droite. La périodicité de B_3 fait qu'on retrouve dans toutes les demi-périodes de gauche la partie de B_1 comprise dans $\left[\nu_1, \nu_2\right]$ et dans

celles de droite la partie de B_1 comprise dans $[-\nu_2, -\nu_1]$ C'est la seule solution pour que la convolution par $R_{1/h}(\nu)$ dans la relation (21) n'introduise pas de distorsion du spectre B_1 initial. Sans ces hypothèses, il se produirait des recouvrements qui changeraient complètement l'allure du spectre.

En résumé, il faut choisir une valeur de h, telle que le domaine $\Delta\nu$ soit inférieur à 1/2 h et s'arranger pour que ce domaine soit entièrement compris dans une des demi-périodes de $B_3(\nu)$. C'est à ce dernier aspect du choix que nous allons revenir maintenant en envisageant les trois cas qui peuvent se présenter.

α) $\Delta\nu = [\nu_1, \nu_2]$ avec $\nu_1 = 0$

(22) Dans ce cas, il suffit de prendre $\frac{1}{2h} = \nu_2$

$$\boxed{h = 1/2\nu_2}$$

β) $\Delta\nu = [\nu_1, \nu_2]$ avec $\nu_1 > 0$ et $\nu_2 = n.\Delta\nu$ (n entier)

Là, il faut prendre : $\frac{1}{2h} = \Delta\nu$ et la $n^{ième}$ demi-période de $B_3(\nu)$ coïncidera avec le domaine $[\nu_1, \nu_2]$ de $B_1(\nu)$

(23) $$\boxed{h = \frac{1}{2\Delta\nu}}$$

γ) $\Delta\nu = [\nu_1, \nu_2]$ avec $\nu_1 > 0$ et $n\Delta\nu \leq \nu_2 \leq (n+1)\,\Delta\nu$

On définit un nouveau domaine $\Delta\nu' \geq \Delta\nu$ tel que :

$\Delta\nu' = \frac{\nu_2}{n}$ et on prend : $\frac{1}{2h} = \Delta\nu'$

(24) $$\boxed{h = \frac{n}{2\nu_2} = \frac{1}{2\Delta\nu'}}$$ qui est l'expression la plus générale pour h.

I.4. - TRANSFORMEE DE FOURIER D'UNE FONCTION LIMITEE ET ECHANTILLONNEE

Le calcul numérique du spectre d'une fonction expérimentale ne peut se faire qu'à partir d'un nombre fini d'échantillons de cette fonction. Nous avons vu successivement quelles étaient les conséquences spectrales d'une

limitation ou d'un échantillonnage infini d'une fonction. Dans la pratique, on conjugue les deux opérations et on est amené à calculer le spectre par l'expression :

$$B_c(\nu) = h \sum_{n=-N}^{N-1} I_1(nh) e^{2i\pi nh\nu} \tag{25}$$

Il nous faut donc étudier maintenant la T.F. de la fonction $I_4(t)$ limitée et échantillonnée définie par :

$$I_4(t) = I_1(t) . D_{2T}(t) . R_h(t) \tag{26}$$

Rappellons que D_{2T} est la fonction "rectangle" définie en (I.2) qui permet de limiter le domaine de $I_1(t)$ à $[-T,+T]$, et que $R_h(t)$ est la distribution de Dirac de période h.

L'expression (26) définit un échantillonnage de N'=2N valeurs (fig. 2,a).

Soit $B_4(\nu)$ la T.F. de $I_4(t)$. (fig. 2,a)

$$B_4(\nu) = T^+[I_4(t)] \tag{27}$$

Dans l'expression (26), on peut regrouper les deux premiers termes ce qui donne :

$$I_4(t) = I_2(t) . R_h(t) \tag{28}$$

Prenons la T.F. de cette expression :

$$B_4(\nu) = B_2(\nu) * R_{1/h}(\nu) \quad \text{au coefficient } 1/h \text{ près} \tag{29}$$

Développons cette expression. Nous obtenons :

$$B_4(\nu) = B_2(\nu) * \left[\sum_{k=-\infty}^{+\infty} \delta\left(\nu - \frac{k}{h}\right) \right] \tag{30}$$

$$B_4(\nu_0) = \int_{-\infty}^{+\infty} B_2(\nu) . \sum_{k=-\infty}^{+\infty} \delta\left(\nu_0 - \nu + \frac{k}{h}\right) d\nu \tag{31}$$

Comme $\delta(\nu_0 - \nu + \frac{k}{h})$ vaut $\begin{cases} 1 \text{ pour } \nu = \nu_0 + \frac{k}{h} \\ 0 \text{ partout ailleurs} \end{cases}$

l'expression (31) devient :

$$B_4(\nu_0) = \sum_{k=-\infty}^{+\infty} B_2(\nu_0 + \frac{k}{h}) . \delta(0)$$

et finalement :

$$(32) \qquad B_4(\nu_0) = B_2(\nu_0) + \sum_{k \neq 0} B_2 (\nu_0 + \frac{k}{h})$$

Si h a la valeur $1/2\Delta\nu$ de l'expression (24) et si ν_0 fait partie du domaine $[-\nu_2, -\nu_1]$ ou $[\nu_1, \nu_2]$, nous voyons que seul le 1er terme de l'expression (32) se trouvera dans ce domaine, $B_4(\nu_0)$ sera différent de $B_2(\nu_0)$ puisque comme on l'a vu en (I.2), la fonction $B_2(\nu_0)$ à résolution finie a un support illimité.

Les valeurs $B_2(\nu_0 + k/h)$ intervenant dans le 2ème terme de (32) sont toutes à l'extérieur des deux domaines $\Delta\nu$ là où justement la fonction $B_2(\nu)$ devient très vite négligeable surtout lorsque la limite de résolution $\delta\nu << \Delta\nu$.

Donc, avec une excellente approximation on peut dire :

$$(33) \qquad B_4(\nu_0) \not\equiv B_2(\nu_0)$$

Et en tout état de cause la différence entre (32) et (33) est définie mathématiquement et aisément mesurable.

I.5. - TRANSFORMEE DE FOURIER DISCRETE NORMALISEE (DFTN)

I.5.1. - Transformée de Fourier discrète - D.F.T.

La fonction $B_4(\nu)$ étudiée au paragraphe précédent est une fonction continue et périodique, mais le résultat d'une transformée de fourier numérique est une série d'échantillons.

D'après (27) nous avons : $B_4(\nu) = T^+ \left[I_4(t)\right]$. Par réciprocité de la transformée nous avons donc :

(34) $\quad I_4(t) = T^- \left[B_4(\nu)\right]$

La fonction échantillonnée $I_4(t)$ est limitée au domaine $\left[-T,+T\right] = \left[-Nh,+Nh\right]$. On peut donc appliquer à $B_4(\nu)$ le théorème d'échantillonnage (I,3(16)) et se contenter de calculer les valeurs $B_4(\nu_k)$ définies pour $\nu_k = k/2T = k\ r$

(35) $\quad$ avec $r = \frac{1}{2T} = \frac{1}{2Nh}$

Soit $B(\nu)$ (fig. 2,b) la fonction constituée par l'ensemble des valeurs $B_4(\nu_k)$. Nous pouvons écrire :

(36) $\quad B(\nu) = B_4(\nu).\ R_r(\nu)$

Où $R_r(\nu)$ est une distribution de Dirac de période r. Soit I(t) la T.F. de $B(\nu)$

(37) $\quad I(t) = T^- \left[B(\nu)\right]$

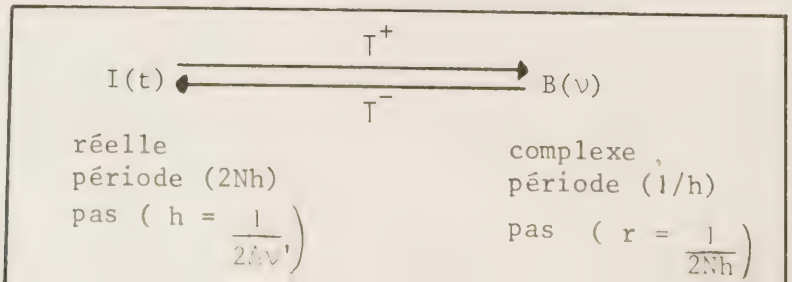
$$I(t) = T^- \left[B_4(\nu).\ R_r(\nu)\right]$$

Et d'après (8) et (35) :

(38) $\quad I(t) = I_4(t) \ast R_{1/r}(t)$ $\quad$ au coefficient 1/r près

I(t) est donc une fonction périodique de période $2T=2Nh$ définie de $-\infty$ à $+\infty$ et identique à $I_4(t)$ dans la période centrée sur l'origine.

EN RESUME : Nous nous trouvons en présence de deux fonctions périodiques échantillonnées qui sont transformées l'une de l'autre (fig. 2,b). Le tableau ci-dessous en résume les caractéristiques.



Voyons maintenant plus en détails comment il est possible de passer de l'une à l'autre de ces deux fonctions par un calcul numérique ?

Nous avons vu en (27) que le spectre continu $B_4(\nu)$ était la transformée de $I_4(t)$ qui est identique à la période de I(t) centrée sur l'origine. Les valeurs $B(\nu_k)$ étant des échantillons de $B_4(\nu)$, leur calcul numérique se fera par l'expression suivante :

$$(42) \qquad B(kr) = h \sum_{n=-N}^{+N-1} I(nh)\, e^{2i\pi \frac{nk}{2N}}$$

Comme I(t) et l'exponentielle sont périodiques de période 2N, nous avons quelque soit l'entier n1 :

$$(43) \qquad B(kr) = h \sum_{n=n1}^{n1+2N-1} I(nh)\, e^{2i\pi \frac{nk}{2N}}$$

Le spectre échantillonné **B** peut donc être calculé par une T.F. discrète à partir d'une séquence quelconque de 2N valeurs de I.

Inversement, I(nh) peut être obtenu par :

$$(44) \qquad I(nh) = r \sum_{q=n2}^{n2+2N-1} B(qr)\, e^{-2i\pi \frac{nq}{2N}} \quad \forall n2 \text{ entier} \geq 0 \text{ ou} < 0$$

En effet, si on porte cette valeur dans (43), on obtient :

$$B'(kr) = hr \sum_{n=n1}^{n1+2N-1} \sum_{q=n2}^{n2+2N-1} B(qr)\, e^{2i\pi m \frac{(k-q)}{2N}}$$

Qui peut s'écrire aussi :

$$(45) \qquad B'(kr) = hr \sum_{q=n2}^{n2+2N-1} B(qr) \sum_{n=n1}^{n1+2N-1} e^{2i\pi n \frac{(k-q)}{2N}}$$

Du fait de l'orthogonalité des fonctions exponentielles, la deuxième somme n'est différente de zéro que si q=k.

En effet : $\sum_{n=n1}^{n1+2N-1} e^{2i\pi n \frac{(k-q)}{2N}} = \begin{cases} 2N & \text{si } q=k \text{ mod. } 2N \\ 0 & \text{dans tous les autres cas} \end{cases}$

L'expression (45) devient alors :

$$B'(kr) = hr.\ B(kr).\ 2N$$

Et comme r = 1/2Nh on obtient bien :

$$B'(kr) = B(kr)$$

EN RESUME : A partir d'une séquence quelconque de 2N valeurs de l'une des deux fonctions I(t) ou B(ν), il est possible de calculer toutes les valeurs de l'autre fonction par l'une des deux relations (43) ou (44).

I.5.2. - Transformée de Fourier discrète normalisée - D.F.T.N.

Les relations (43) et (44) sont dépendantes des unités t et ν par l'intermédiaire des constantes h et r. On en simplifie l'écriture, en adoptant les conventions normalisées suivantes :

$$\begin{cases} 2T = 2Nh = 1 \ ; \ n1=n2=0 \\ \text{et } N' = 2N \end{cases}$$

ce qui implique : $h = \frac{1}{N'}$; $r = \frac{1}{N'h} = 1$; $2\Delta\nu'=N'= \frac{1}{h}$

on pose aussi $\begin{cases} W_{N'} = e^{+2i\pi/N'} \\ B(kr) = B_k \\ I(nh) = I_n \end{cases}$

Les relations (43) et (44) s'écrivent alors:

(46) $$B_k = \frac{1}{N'} \sum_{n=0}^{N'-1} I_n W_{N'}^{+nk}$$

(47) $$I_n = \sum_{q=0}^{N'-1} B_q W_{N'}^{-nq}$$

Définition de la "D.F.T.N." (N')

{h = $\frac{1}{N'}$, r = 1}

La plupart des programmes dits de "Transformée de Fourier" quand ils utilisent la méthode de calcul rapide "F.F.T." de Cooley Tuckey calculent en fait la "D.F.T.N." des N' échantillons de départ : I_n pour n = 0,1,2,...,N'-1.

En sortie, le programme fournit les N' valeurs du spectre B_k pour k = 0,1,2,...,N'-1.

REMARQUES :

α) IMPORTANCE DE L'EMPLACEMENT DES ECHANTILLONS.

Les valeurs B_k calculées par une DFTN ne seront égales aux valeurs B (kr) calculées par (43) que si l'abcisse du premier échantillon de la fonction à transformer se trouve bien à l'origine. En particulier si on dispose de l'échantillonnage :

$$\{I(-Nh), I(-(N-1)h), \ldots, I(0), I(h), \ldots I((N-1)h)\}$$

Il faut tenir compte de la périodicité de la fonction I(t) et décaler l'échantillonnage pour qu'il parte bien de l'origine en prenant :

$$\left\{ \begin{array}{l} I_0 = I(0) \; ; \; I_1 = I(h) \; ; \ldots\ldots ; \; I_{N-1} = I((N-1)h) \; ; \\ \\ I_N = I(-Nh) \; ; \; I_{N+1} = I(-(N-1)h) \; ; \ldots . ; \; I_{N'} = I(-h) \end{array} \right\}$$

Dans ce cas particulier, il y aurait un changement de signe des valeurs B_k d'indice impair, si on ne prenait pas cette précaution.

Dans les cas où il n'est pas possible de partir d'un échantillon sur l'origine, l'application de la DFTN fait subir l'équivalent d'une translation à la fonction à transformer.

Or on sait que : si $\left\{ \begin{array}{l} T^+ \left[I(t) \right] = B(\nu) \\ \\ T^+ \left[I(t-t') \right] = B(\nu)\, e^{2i\pi\nu t'} \end{array} \right.$

(48) alors

Dans ce cas, seul le module du spectre complexe ainsi calculé aura une signification et on l'obtiendra par :

$$(49) \qquad |B_k| = (B_{pk}^2 + B_{ik}^2)^{1/2}$$

où $\left\{ \begin{array}{l} B_{pk} = \text{partie réelle et paire de } B_k \\ B_{ik} = \text{partie imaginaire et impaire de } B_k \end{array} \right.$ (voir (I,2(3))

On retrouve alors tous les inconvénients de non linéarité et de déformation de la fonction d'appareil liés à ce type de calcul [3].

β) NORMALISATION DU PAS D'ECHANTILLONNAGE.

Le fait que h doit être égal à 1/N' n'est pas génant puisqu'il est toujours possible de faire subir à I(t) une homothétie pour que cette condition soit réalisée.

Or on sait que : si $B(\nu) = T^+\left[I(t)\right]$

(50) alors $T^+\left[I(kt)\right] = \frac{1}{|k|} B\ (t/k)$

si $k \neq 0$

Le spectre subira une homothétie inverse, mais à une constante multiplicative près, ses valeurs ne seront pas modifiées.

D'un point de vue pratique, disons que si on applique une DFTN aux N' valeurs d'une fonction réelle échantillonnée avec un pas h à partir de l'origine, les N' valeurs B_k du spectre ainsi calculées seront réparties sur une période de largeur 1/h, le pas d'échantillonnage du spectre étant alors égal à $r = \frac{1}{N'h}$. L'abcisse ν_1 du 1er échantillon sera de la forme : $\nu_1 = k/h$. Une localisation approximative et a priori des fréquences ν du spectre recherché est donc nécessaire pour déterminer la valeur entière k.

Dans le cas de la figure (2,b), c'est le triangle hachuré qui correspond au spectre cherché et on a k = 1.

II. - UTILISATION DE L'ALGORITHME F.F.T. POUR LE CALCUL DE LA D.F.T.N. D'UNE FONCTION REELLE IMPAIRE.

II.1. - ETUDE THEORIQUE.

II.1.1. - Définition de la F.F.T. par la technique dite de "decimation in time".

Partons de la "D.F.T.N." définie en (I.5.2). Nous avions :

$$B_k = \frac{1}{N'} \sum_{n=0}^{N'-1} I_n W_{N'}^{+nk} \qquad (51)$$

pour $k=0,1,\ldots,N'-1$ avec $W_{N'} = e^{+2i\pi/N'}$

Soit $\{B_k\}_{N'}$ l'ensemble des N' valeurs complexes B_k

Soit $\{I_n\}_{N'}$ l'ensemble des N' valeurs complexes I_n

(le cas où I_n est réel, est en effet un cas particulier)

$$\{I_n\}_{N'} \underset{\text{DFTN}^-}{\overset{\text{DFTN}^+}{\rightleftarrows}} \{B_k\}_{N'}$$

(le signe - indique que l'exponentielle est négative)

Le calcul de $\{B_k\}_{N'}$ à partir de $\{I_n\}_{N'}$ par l'équation (51) nécessite N'^2 opérations complexes. Une application directe de cette formule dans un programme conduirait dès que N' est grand à des temps de calcul prohibitifs. Des algorithmes généralement connus sous le sigle "F.F.T." (fast fourier transform) ont été mis au point ces dernières années à partir des travaux de Cooley et Tuckey [17,18 à 33] sur ce sujet. Le propre de ces algorithmes est de rendre le temps de calcul non plus proportionnel à N'^2 mais proportionnel à $N' \log_2 N'$. Ils supposent en général que N' est une puissance de 2. La technique dite de "Decimation in Time" que nous allons rappeler est particulièrement bien adaptée [34] à l'élaboration d'un programme de T.F. rapide.

Montrons d'abord que la $D.F.T.N._{(N')}$ de n'importe quelle fonction complexe est une combinaison linéaire de la $D.F.T.N._{(N'/2)}$ de deux fonctions ayant moitié moins de points et issues de la première.

Soit $\{Y_n\}_{N'/2}$ et $\{Z_n\}_{N'/2}$ les ensembles de N'/2 valeurs correspondant aux valeurs paires et impaires de l'indice n.

$$(52)\qquad I_n \quad n \in (0,1,2,\ldots,N'-1) \rightarrow \begin{cases} Y_n = I_{2n} \\ Z_n = I_{2n+1} \end{cases} \qquad n \in (0,1,2,\ldots,N'/2-1)$$

Soit $\{C_k\}_{N'/2}$ et $\{D_k\}_{N'/2}$ la D.F.T.N.$^{+}_{(N'/2)}$ de $\{Y_n\}_{N'/2}$ et $\{Z_n\}_{N'/2}$

$$(53)\qquad \begin{cases} C_k = \sum_{n=0}^{N'/2-1} Y_n W_{N'/2}^{nk} = \sum_{n=0}^{N'/2-1} Y_n W_{N'}^{2nk} \\ D_k = \sum_{n=0}^{N'/2-1} Z_n W_{N'}^{2nk} \end{cases} \qquad \text{pour } k = 0,1,2,\ldots,N'/2-1$$

Dans (51), nous allons séparer les termes pairs et impairs ce qui donne :

$$(54)\qquad B_k = \sum_{n=0}^{N'/2-1} Y_n W_{N'}^{2nk} + \sum_{n=0}^{N'/2-1} Z_n W_{N'}^{(2n+1)k}$$

Et en tenant compte de (53) :

$$B_k = C_k + W_{N'}^{k} \cdot D_k$$

Comme nous l'avons vu à propos des relations (43) et (44), C_k et D_k sont périodiques de période N'/2.

$$C_{k+N'/2} = C_k \quad \text{et} \quad D_{k+N'/2} = D_k$$

D'autre part, $W_{N'}^{k+N'/2} = -W_{N'}^{k} = -e^{2i\pi\frac{k}{N'}}$

D'où les relations importantes :

$$(55)\qquad \boxed{\begin{aligned} B_k &= C_k + W_{N'}^{k} D_k \\ B_{k+N'/2} &= C_k - W_{N'}^{k} D_k \qquad \text{avec } k = 0,1,\ldots,N'/2-1 \end{aligned}}$$

Il s'ensuit qu'il est possible de calculer la D.F.T.N. de $\{I_n\}_{N'}$ à partir de la D.F.T.N. de $\{C_k\}_{N'/2}$ et $\{D_k\}_{N'/2}$. La figure 3 illustre le calcul normal d'une D.F.T.N. avec N' = 8. La figure 4 illustre le même calcul mais à partir des D.F.T.N. de $\{C_k\}_{N'/2}$ et $\{D_k\}_{N'/2}$.

Le même procéssus peut être appliqué pour le calcul de $\{C_k\}_{N'/2}$ et $\{D_k\}_{N'/2}$. On arrive ainsi à l'organigramme décrit sur la figure 5. Cet organigramme peut être généralisé à une valeur quelconque de N' si N' est une puissance de 2.

L'application des relations (55) divise par 2 le nombre de points auxquels il faudra appliquer la D.F.T.N. et introduit une étape d'intercombinaison.

Si $N' = 2^n$ ($n = \log_2 N'$), il faudra n étapes d'intercombinaisons et dans chaque étape le nombre d'opérations complexes sera proportionnel à N'. On voit donc bien apparaitre le fait qu'un calcul de D.F.T.N. par un algorithme de ce type est proportionnel à N'.n soit $N' \log_2 N'$.

En regardant la figure 5, on voit aussi qu'il sera nécessaire de prévoir un triage des échantillons I_n avant d'appliquer l'algorithme. C'est aussi un des éléments caractéristiques de ces méthodes. Ce triage peut être fait par des techniques de triage proprement dit ou bien en utilisant les propriétés dites "d'inversion de bits"[24][35] que nous ne développerons pas ici.

Un autre avantage très important de cet algorithme est qu'en diminuant le nombre d'opérations, on augmente la précision du résultat.

II.1.2. - Adaptation de l'algorithme "F.F.T." au cas de la transformée d'une fonction réelle.

La "D.F.T.N." est définie par l'expression :

$$B_k = \frac{1}{N'} \sum_{n=0}^{N'-1} I_n W_{N'}^{+nk} \qquad k = 0,1,2,\ldots,N'-1$$

On voit que tout programme qui utilise l'algorithme "F.F.T." défini précédemment à partir de cette expression a besoin d'un tableau de 2N' mots

en mémoire centrale pour stocker :

en entrée, les N' valeurs complexes de I_n

en sortie, les N' valeurs complexes de B_k

Rappelons que N' est la puissance de 2 qui est immédiatement supérieure ou égale au nombre de valeurs connues de I_n. Ces valeurs doivent donc être complétées avec des valeurs nulles pour constituer l'ensemble $\{I_n\}_{N'}$.

Si la fonction à transformer est réelle, il faut mettre à zéro les N' valeurs imaginaires en entrée. D'autre part, en sortie, seules N'/2 valeurs complexes du spectre sont utiles puisque :

$$\{I_n\}_{N'} \text{ réel} \Longrightarrow \begin{cases} \{B_k\} \quad \text{hermitique} \\ B_k = B^*_{-k} = B^*_{N'-k} \end{cases} \qquad \text{(fig. 2,b)}$$

Bien entendu, l'algorithme général est applicable à une fonction réelle, mais il conduit à un gaspillage de place mémoire et de temps de calcul. Nous allons donc définir un algorithme adapté aux fonctions réelles pour gagner un facteur 2 sur l'encombrement en mémoire centrale et sur le temps de calcul.

Soit $\{E_n\}_N$ l'ensemble des N = N'/2 valeurs complexes définies à partir de $\{I_n\}_{N'}$ par :

(56) $$E_n = Y_n + i\, Z_n \qquad \text{pour } n = 0,1,2,\ldots,N-1$$

$$\text{avec} \begin{cases} Y_n = I_{2n} \\ Z_n = I_{2n+1} \end{cases} \qquad \text{(voir (52))}$$

(57) $$\begin{cases} \{C_k\}_N = \text{DFTN}^+ \{Y_n\}_N \\ \{D_k\}_N = \text{DFTN}^+ \{Z_n\}_N \end{cases}$$

Soit $\{F_k\}_N$ l'ensemble transformé de $\{E_n\}_N$

(58) $$\{F_k\}_N = \text{DFTN}^+ \{E_n\}_N$$

(59) $F_k = C_k + i\, D_k$ d'après (56) et (57)

Comme $\{Y_n\}_N$ et $\{Z_n\}_N$ sont réels, nous avons :

$$\begin{cases} C_k = C^*_{-k} = C^*_{N-k} \\ D_k = D^*_{-k} = D^*_{N-k} \end{cases}$$

Et finallement :

(60) $F^*_{N-k} = C^*_{N-k} - i\, D^*_{N-k} = C_k - i\, D_k$

Des relations (59) et (60) on tire alors :

(61) $C_k = \dfrac{F_k + F^*_{N-k}}{2}$ et $D_k = \dfrac{F_k - F^*_{N-k}}{2i}$

D'autre part, nous avons vu en (55) que si $\{B_k\}_{N'} = DFTN^{\pm}\{I_n\}_{N'}$

alors : $\begin{cases} B_k = C_k + W^{\pm k}_{N'}\, D_k \\ B_{N+k} = C_k - W^{\pm k}_{N'}\, D_k \end{cases}$ avec $W^k_{N'} = e^{2i\pi \frac{k}{N'}}$

D'où finallement :

(62)
$$\begin{cases} B_k = \dfrac{F_k + F^*_{N-k}}{2} + W^{\pm k}_{N'}\, \dfrac{F_k - F^*_{N-k}}{2i} & \text{Si } \{F_k\}_N = DFTN^{\pm}_{(N)}\{E_n\}_N \\ B_{N+k} = \dfrac{F_k + F^*_{N-k}}{2} - W^{\pm k}_{N'}\, \dfrac{F_k - F^*_{N-k}}{2i} & \text{avec } k = 0,1,2,..,N-1 \end{cases}$$

En fin de compte, on est amené à remplacer l'opération (I) par l'ensemble des opérations schématisées par (II)

(I) $\{I_n\}_{N'} \xrightarrow{DFTN^{\pm}_{(N')}} \{B_k\}_{N'}$

(II)

$$\{I_n\}_{N'=2N} \xrightarrow{(56)} \{E_n\}_N \xrightarrow{DFTN^{\pm}_{(N)}} \{F_k\}_N \xrightarrow{(62)} \{B_k\}_{N'}$$

réel — complexe — complexe — complexe hermitique

Le gain de temps concernant la DFTN proprement dite est égal

à : $\dfrac{N' \log_2 N'}{N \log_2 N} = 2 \times \dfrac{n'}{n'-1}$ si $n' = \log_2 N'$

Il est donc supérieur à 2 mais il faut aussi réaliser les opérations supplémentaires (56) et (62) ce qui ramène le gain global au voisinage de 2.

II.1.3. - ADAPTATION DE L'ALGORITHME "FFT" AU CAS DE LA TRANSFORMEE D'UNE FONCTION RELLE IMPAIRE (OU PAIRE).

Dans ce paragraphe, nous n'envisageons que le cas d'une fonction impaire, mais l'ensemble du raisonnement est semblable dans le cas d'une fonction paire.

Il est facile de particulariser le cas de la fonction réelle impaire à partir de celui d'une fonction réelle quelconque I(t) en constatant que cette dernière est toujours la somme d'une fonction paire $I_p(t)$ et d'une fonction impaire $I_i(t)$ (fig. 2,b,c,d).

Ces fonctions peuvent être calculées par l'expression (63) :

(63)
$$\begin{cases} I_p(nh) = \frac{1}{2}\,(I(nh) + I(-nh)) \\ I_i(nh) = \frac{1}{2}\,(I(nh) - I(-nh)) \end{cases}$$

(64) $$I(t) = I_p(t) + I_i(t)$$

Nous avons vu en (I.5.1) que :

(65) $$B(\nu) = T^+\left[I(t)\right] = B_p(\nu) + i\,B_i(\nu)$$

$B_p(\nu)$ et $B_i(\nu)$ sont respectivement des fonctions réelles paires et impaires

$$(66) \qquad T^+\left[I(t)\right] = T^+\left[I_p(t) + I_i(t)\right] = T^+\left[I_p(t)\right] + T^+\left[I_i(t)\right]$$

$$= T_c\left[I_p(t)\right] + iT_s\left[I_i(t)\right] \quad \text{d'après (3)}$$

En comparant (65) et (66), nous avons :

$$(67) \qquad \begin{cases} T_c\left[I_p(t)\right] = B_p(\nu) \\ \\ T_s\left[I_i(t)\right] = B_i(\nu) \end{cases} \qquad \text{(fig. 2, d et c)}$$

Donc si $I(t) = I_i(t)$, c'est-à-dire si <u>I(t) est impaire</u>, nous aurons :

$$(68) \qquad T^+\left[I(t)\right] = iT_s\left[I(t)\right] = i\,B_i(\nu)$$

Comme nous l'avons vu à propos de la transformée de Fourier discrète, nous nous trouvons en présence de deux fonctions périodiques échantillonnées qui sont transformées l'une de l'autre. A partir d'une séquence quelconque de 2N valeurs de I(t), il est possible de calculer les valeurs de $B_i(\nu)$ par la relation (43) qui peut s'écrire :

$$(69) \qquad B_i(kr) = h \sum_{n=0}^{2N-1} I(nh)\, e^{2i\pi \frac{nk}{2N}}$$

Mais d'après (68), (69) peut s'écrire :

$$(70) \qquad B_i(kr) = h \sum_{n=0}^{2N-1} I(nh) \sin\left(2\pi \frac{nk}{2N}\right)$$

Comme I(nh) est impaire et périodique avec 2N échantillons par période, elle est impaire par rapport à l'abcisse Nh. En effet :

$$I\,(Nh - nh) = -\,I\,(-Nh + nh) = -\,I\,(Nh + nh)$$

D'où :

$$(71) \qquad \boxed{B_i(kr) = 2\,h \sum_{n=0}^{N-1} I(nh) \sin\left(2\pi \frac{nk}{2N}\right)}$$

Nous voyons donc que dans ce cas particulier, l'ensemble

des deux fonctions périodiques $I(t) = I_i(t)$ et $B_i(\nu)$ sont complètement déterminées si on en connait N valeurs partant de l'origine. I(t) peut être considéré comme la somme de sinusoïdes ayant toutes la phase nulle à l'origine. C'est cette propriété qui permet de rattacher le calcul d'une D.F.T. par l'expression (71) au cas plus général d'une détection synchrone.

Comparons (71) à l'expression de la D.F.T.N. d'une fonction complexe I(t).

$$B_k = \frac{1}{2N} \sum_{n=0}^{2N-1} I_n \, e^{2i\pi \frac{nk}{2N}} \qquad (72)$$

Nous avons vu qu'un algorithme développé à partir de (72) nécessitait 4N mots en mémoire centrale et $2N \log_2 2N$ opérations complexes. Nous avons montré que dans le cas d'une fonction I(t) réelle on pouvait gagner un facteur 2 sur l'encombrement en mémoire et sur la durée de calcul. Dans le cas d'une fonction réelle impaire, l'expression (71) montre que N mots en mémoire centrale doivent être suffisants et nous allons montrer que, par rapport au cas de la fonction réelle (II.1.2), il est encore possible de gagner un facteur de l'ordre de 2 sur le temps de calcul.

REMARQUE :

Si on veut appliquer aux N = N'/2 valeurs, $I_0, I_1, \ldots, I_{N-1}$ d'une fonction réelle impaire un programme de D.F.T.N. prévu pour une fonction complexe, il faut prendre la précaution d'ajouter à ces N valeurs, N zéros pour constituer N'valeurs réelles en entrée. (Il faut aussi mettre à zéro les N' valeurs imaginaires). Sinon, étant donné le caractère figé de la D.F.T.N., les N valeurs du spectre calculées sur une période $2\Delta\nu' = 1/h = N'$ (voir I.5.2. (46)) seraient distantes de $r' = \frac{N'}{N} = 2$ au lieu de r = 1 et il y aurait alors sous échantillonnage et perte d'information au sens du théorème d'échantillonnage (I.3.).

Avant de définir l'algorithme F.F.T. adapté au cas d'une fonction impaire, nous allons montrer qu'il est possible de calculer les N' valeurs réelles de la D.F.T.N. d'une fonction hermitique de N' valeurs à l'aide d'une $D.F.T.N._{(N)}$ ne portant que sur N = N'/2 valeurs. Pour le faire, nous allons reprendre le raisonnement de II.1.2 mais en l'inversant.

D.F.T.N. d'une fonction hermitique

Soit $\{B_k\}_{N'}$, un ensemble de N' valeurs d'une fonction complexe hermitique $B(\nu)$ et soit $\{I_n\}_{N'}$, l'ensemble des N' valeurs qui en est la transformée :

$$\{B_k\}_{N'} \xrightarrow{DFTN^{\pm}_{(N')}} \{I_n\}_{N'} \tag{73}$$

complexe hermitique — réel

A partir des relations (62), nous pouvons définir $\{F_k\}_N$ par :

$$\begin{cases} F_k + F^*_{N-k} = B_k + B_{N+k} \\ F_k - F^*_{N-k} = i\ W^{\pm k}_{N'} \left[B_k - B_{N+k}\right] \end{cases} \tag{74}$$

$$F_k = \frac{1}{2}\left[B_k + B_{N+k} + i\ W^{\pm k}_{N'}\ (B_k - B_{N+k})\right] \tag{75}$$

or $B_{N+k} = B^*_{-N-k}$ puisque B est hermitique

et $B^*_{-N-k} = B^*_{N-k}$ puisque B est périodique de période 2N

D'où : $B_{N+k} = B^*_{N-k}$

L'expression (75) s'écrit alors :

$$F_k = \frac{1}{2}\left[B_k + B^*_{N-k} + i\ W^{\pm k}_{N'}\ (B_k - B^*_{N-k})\right] \tag{76}$$

Soit $\{E_n\}_N$ la D.F.T.N. de $\{F_k\}_N$

$$\{E_n\}_N \xrightarrow{DFTN^{\pm}_{(N)}} \{F_k\}_N$$

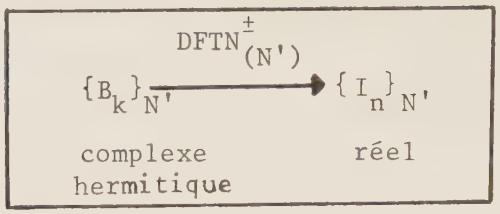
Comme dans la relation (56), on peut considérer que chaque valeur complexe E_n peut s'écrire sous la forme :

$$E_n = I_{2n} + i\ I_{2n+1} \tag{77}$$

ce qui permet de définir l'ensemble $\{I_n\}_{N'}$, cherché.

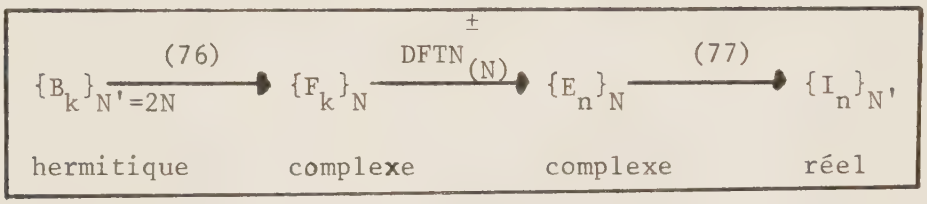
Les schémas qui suivent résument toutes ces opérations :

Le Schéma (I)



peut être remplacé par le schéma (III) suivant :

(III)



D.F.T.N. d'une fonction réelle impaire I(t)

I(t) étant une fonction échantillonnée impaire et périodique avec 2N valeurs par période (fig. 2,d), nous avons vu au début de ce paragraphe, qu'il est possible de la transformer par une D.F.T.N.$_{(N')}$ soit à partir d'une séquence $\{I_n\}_{N'}$ de 2N valeurs, soit à partir d'une séquence $\{I_n\}_N$ de N valeurs partant de l'origine et complétées par N valeurs nulles.

(78) $\left.\begin{matrix}\{I_n\}_{N'} \\ \{I_n\}_N\end{matrix}\right\} \xrightarrow{DFTN^{\pm}_{(N')}} \{B_k\}_{N'}$

Pour éviter le gaspillage de place mémoire et de temps calcul qu'entraine cette opération, nous allons définir un nouvel algorithme. A partir des N valeurs $\{I_n\}_N$ qui partent du point de symétrie, nous définissons un ensemble $\{H_n\}_N$ de N valeurs complexes d'une fonction hermitique et périodique H(t) dont la DFTN$_{(N)}$ calculée d'après le schéma (III) sera un ensemble $\{K_k\}_N$ de N valeurs réelles. Les valeurs B_k cherchées seront alors calculées à partir d'une combinaison des valeurs K_k.

I(t) étant impaire et périodique, nous avons :

(78) $I_n = - I_{2N-n}$ ou $I_{N+n} = - I_{-N-n} = - I_{N-n}$

Soit $\{G_n\}_N$ l'ensemble des N valeurs complexes définies par :

(79) $$G_n = I_{2n+1} + i\, I_{2n} \quad ((*)) \qquad \text{pour } n = 0,1,2,\ldots,N-1$$

Nous définissons alors à partir de $\{G_n\}_N$ un ensemble $\{H_n\}_N$ défini par :

(80) $$H_n = \frac{1}{2}\,(G_n + G^*_{N-n})$$

Mais, $G^*_{N-n} = I_{2(N-n)+1} - i\, I_{2(N-n)}$

$$G^*_{N-n} = -I_{2n-1} + i\, I_{2n} \qquad \text{d'après (78)}$$

En portant ces valeurs dans H_n, on trouve :

$$H_n = \frac{1}{2}\left[I_{2n+1} + i\, I_{2n} - I_{2n-1} + i\, I_{2n}\right]$$

(81) $$\boxed{H_n = \frac{1}{2}\left[I_{2n+1} - I_{2n-1}\right] + i\, I_{2n}} \qquad n = 0,1,2,\ldots,N-1$$

On définit ainsi une fonction H(t) hermitique et périodique, chacun de ses périodes étant constituée par l'ensemble $\{H_n\}_N$. Pour montrer que H(t) est hermitique, il suffit de montrer que :

(82) $$H^*_{\frac{N}{2}+p} = H_{-\frac{N}{2}-p} = H_{\frac{N}{2}-p}$$

Or d'après (80), nous avons :

$$\begin{cases} H^*_{\frac{N}{2}+p} = \frac{1}{2}\left(G^*_{\frac{N}{2}+p} + G_{N-\frac{N}{2}-p}\right) = \frac{1}{2}\left(G^*_{\frac{N}{2}+p} + G_{\frac{N}{2}-p}\right) \\ H_{\frac{N}{2}-p} = \frac{1}{2}\left(G_{\frac{N}{2}-p} + G^*_{N-\frac{N}{2}+p}\right) = \frac{1}{2}\left(G_{\frac{N}{2}-p} + G^*_{\frac{N}{2}+p}\right) \end{cases}$$

L'égalité (82) est donc bien vérifiée.

Soit $\{K_k\}_N$ l'ensemble de N valeurs calculé comme DFTN de $\{H_n\}_N$

(83) $$\{H_n\}_N \xrightarrow{\text{DFTN}^+_{(N)}} \{K_k\}_N$$

hermitique réel

((*)) Dans le cas d'une fonction paire on poserait $G_n = I_{2n} + i\, I_{2n+1}$ (pour que H ne soit

Nous avons vu précédemment que ce calcul pouvait être fait à l'aide d'une D.F.T.N$_{(N/2)}$ et des opérations intermédiaires représentées par les équations (76) et (77) (Voir schéma III).

D'après (83), nous avons :

$$K_k = \sum_{n=0}^{N-1} H_n \, W_N^{+kn} = \sum_{n=0}^{N-1} H_n \, W_{N'}^{+2kn}$$

En remplaçant H_n par sa valeur tirée de (81), on obtient :

$$K_k = \frac{1}{2} \sum_{n=0}^{N-1} (I_{2n+1} - I_{2n-1}) \, W_{N'}^{+2kn} + i \sum_{n=0}^{N-1} I_{2n} \, W_{N'}^{+2kn} \tag{84}$$

Considérons alors le terme K'_k défini par :

$$K'_k = \sum_{n=0}^{N-1} I_{2n-1} \, W_{N'}^{+2kn}$$

et faisons le changement de variable $l = n-1$

$$K'_k = \sum_{l=-1}^{N-2} I_{2l+1} \, W_{N'}^{+2k(l+1)}$$

Comme I_{2l+1} et $W_{N'}^{2kl}$ sont périodiques de période N, vis à vis de l, la somme finie sur N valeurs ne change pas si on fait subir une translation à ses limites. D'où :

$$K'_k = \sum_{l=0}^{N-1} I_{2l+1} \, W_{N'}^{+2kl} \cdot W_{N'}^{+2k}$$

L'expression (84) devient alors :

$$K_k = \frac{1}{2} (1 - W_{N'}^{+2k}) \sum_{n=0}^{N-1} I_{2n+1} \, W_{N'}^{+2kn} + i \sum_{n=0}^{N-1} I_{2n} \, W_{N'}^{+2kn} \tag{85}$$

Nous aurions de même :

$$K_{N-k} = \frac{1}{2} (1 - W_{N'}^{-2k}) \sum_{n=0}^{N-1} I_{2n+1} \, W_{N'}^{-2kn} + i \sum_{n=0}^{N-1} I_{2n} \, W_{N'}^{-2kn} \tag{86}$$

Les valeurs K_k ayant été calculées comme DFTN d'une fonction hermitique, sont réelles et on a :

$$K_{N-k} = K^*_{N-k}$$

D'où :

$$K_{N-k} = \frac{1}{2}\left(1 - W_{N'}^{+2k}\right) \sum_{n=0}^{N-1} I_{2n+1}\, W_{N'}^{+2kn} - i \sum_{n=0}^{N-1} I_{2n}\, W_{N'}^{+2kn} \tag{87}$$

Par ailleurs, nous avons d'après (78) :

$$\{B_k\}_{N'} = DFTN^{+}_{(N')} \{I_n\}_{N'} \quad \text{(ou N si on rajoute N zéros)}$$

$$B_k = \sum_{n=0}^{2N-1} I_n\, W_{N'}^{+kn} \qquad \text{qui peut s'écrire :} \tag{88}$$

$$B_k = \sum_{n=0}^{N-1} I_{2n+1}\, W_{N'}^{+2kn} \cdot W_{N'}^{+k} + \sum_{n=0}^{N-1} I_{2n}\, W_{N'}^{+2kn} \tag{89}$$

A partir des relations (85) et (87), nous avons alors :

$$\begin{cases} K_k + K_{N-k} = \left(1 - W_{N'}^{+2k}\right) \sum_{n=0}^{N-1} I_{2n+1}\, W_{N'}^{+2kn} \\ K_k - K_{N-k} = 2i \sum_{n=0}^{N-1} I_{2n}\, W_{N'}^{+2kn} \end{cases}$$

En tenant compte de ces deux égalités, (89) devient :

$$B_k = W_N^{+k} \frac{K_k + K_{N-k}}{1 - W_{N'}^{+2k}} + \frac{1}{2i}\left(K_k - K_{N-k}\right)$$

$$B_k = \frac{K_k + K_{N-k}}{W^{-k} - W^{+k}} - \frac{i}{2}\left(K_k - K_{N-k}\right)$$

et finallement, comme $W_{N'} = e^{\frac{2i\pi}{N'}}$

$$\boxed{B_k = \frac{i}{2}\left[\frac{K_k + K_{N-k}}{\sin \frac{\pi k}{N}} - K_k + K_{N-k}\right]} \qquad k = 0,1,2,\ldots,N'-1 \tag{90}$$

Le schéma qui suit, résume les différentes opérations qui définissent l'algorithme F.F.T. adapté à la transformation d'une fonction réelle impaire.

On remplace (I) par (IV).

(I)

$\{I_n\}_{N'} \xrightarrow{DFTN^{\pm}(N')} \{B_k\}_{N'}$

réel impair — réel impair

(IV)

$\{I_n\}_N \xrightarrow{(81)} \{H_n\}_N \xrightarrow{(76)} \{F_n\}_{\frac{N}{2}} \xrightarrow{DFTN^{\pm}(N/2)} \{E_k\}_{\frac{N}{2}} \xrightarrow{(77)} \{K_k\}_N \xrightarrow{(90)} \{B_k\}_{N'}$

réel impair — hermitique — complexe — complexe — réel — imaginaire impair

II.2. - PROGRAMMATION DE LA F.F.T.

II.2.1. - Transformation d'une fonction complexe en mémoire centrale - Programme "FFT32T".

Il s'agit d'un programme qui calcule la DFTN définie en I.5.2. L'ensemble $\{I_n\}_{N'}$, des N' échantillons complexes d'une fonction complexe I(t) est transformé en un ensemble $\{B_k\}_{N'}$, de N' valeurs complexes de son spectre B(ν).

$$\{I_n\}_{N'} \xrightarrow{DFTN^{\pm}} \{B_k\}_{N'}$$

avec $B_k = \sum_{n=0}^{N'-1} I_n W_{N'}^{nk}$

N' est obligatoirement une puissance de 2, et si le nombre d'échantillons dont on dispose est inférieur, on complète ces échantillons par des valeurs nulles jusqu'à N' (si la valeur moyenne est nulle).

Ce programme réalise toutes les opérations schématisées sur la figure 5. Il se décompose en deux parties distinctes. Dans la première, il réalise le tri des valeurs I_n, dans la deuxième, il effectue les n étapes

d'intercombinaisons. ($n = \log_2 N'$). Pour stocker les N' valeurs complexes I_n, il dispose de deux tableaux T_R et T_I de N' mots en mémoire centrale. Le tableau T_R contient, en entrée, la partie réelle de $\{I_n\}_{N'}$, et en sortie, la partie réelle de $\{B_k\}_{N'}$,. Le tableau T_I contient les parties imaginaires.

II.2.1. α Première partie - Le tri initial des données.

L'ordre dans lequel doivent être rangées les valeurs I_n avant les étapes d'intercombinaison est donné par la règle dite "d'inversion de bits" (B.I.) que nous allons rappeler.

Pour trouver l'indice final m d'une valeur dont l'indice initial n est compris entre 0 et N'-1, il suffit d'inverser sa représentation binaire sur $\log_2 N'$ bits.

exemple : N'=8

n	B.I.	m
0 = (000)	⟶	(000) = 0
1 = (001)	⟶	(100) = 4
2 = (010)	⟶	(010) = 2
3 = (011)	⟶	(110) = 6
4 = (100)	⟶	(001) = 1
5 = (101)	⟶	(101) = 5
6 = (110)	⟶	(011) = 3
7 = (111)	⟶	(111) = 7

$n = 0,1,2,\ldots,N'-1$

De façon générale m sera défini par l'expression :

$$m = I_{N'}(n)$$

Dans le programme "FFT32T", nous avons réalisé cette correspondance entre indices n et m en utilisant la formule de récurrence suivante :

(91)

si $2^p < n \leq 2^{p+1}$

alors $I_N(n) = I_N(n-2^p) + \frac{N}{2^{p+1}}$

avec $I_N(1) = 1$

$n = 1,2,3,\ldots,N$

Si nous appliquons cette formule au cas N'=8, nous obtiendrons :

$I_8(2) = I_8(1) + \frac{N'}{2} = 5$

$I_8(3) = I_8(1) + \frac{N'}{4} = 3$

$I_8(4) = I_8(2) + \frac{N'}{4} = 7$

$I_8(5) = I_8(1) + \frac{N'}{8} = 2$

$I_8(6) = I_8(2) + N'/8 = 6$

$I_8(7) = I_8(3) + N'/8 = 4$

$I_8(8) = I_8(4) + N'/8 = 8$

REMARQUE 1 :

Un procédé pratique pour retrouver dans un cas simple (N petit) cette correspondance entre indices, est schématisé sur la figure 6. Il nous est d'ailleurs suggéré par la méthode de "decimation in time" (II.1.1). A partir des N' valeurs de la série A, on constitue deux autres séries B et C de N'/2 valeurs en prenant successivement une valeur sur deux. Pour la première série B on part de la première valeur de A et pour la deuxième série C on part de la deuxième valeur de A. On recommence ainsi $\log_2 (N'/2)$ fois jusqu'à ce que les dernières séries ainsi définies n'aient plus que deux valeurs.

REMARQUE 2 :

Reportons nous encore au schéma de la figure 6. Si, après le tri, on regroupe les indices en quatre ensembles S_j (avec j=1,2,3,4) de N'/4 indices, nous constatons trois particularités intéressantes :

- a - L'indice initial de chaque sous groupe S_j est égal à $I_4(j)$.

- b - Si dans la série A de départ, on prend un indice sur quatre à partir de $I_4(j)$, on retrouve tous les indices qui font partie de l'ensemble S_j, mais il reste à les ordonner.

- c - Pour ordonner un ensemble S_j où les N'/4 indices ont été rangés par ordre croissant de valeur, il suffit de lui appliquer les correspondances d'indices définies par $I_{N'/4}(n)$. C'est-à-dire que le $n^{ième}$ indice de l'ensemble S_j sera mis en $m^{ième}$ position avec $m=I_{N'/4}(n)$.

Bien entendu les 3 propriétés a, b et c sont valables quelque soit le nombre pair de sous ensembles S_j que l'on désire constituer.

Application de la remarque 2 au programme :

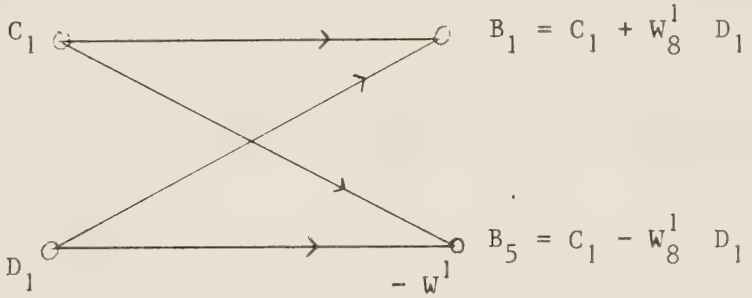
Les N' valeurs des tableaux T_R et T_I seront triées par groupe de N'/4 valeurs. Nous avons donc réservé un tableau supplémentaire S de N'/2 mots en mémoire centrale. Les N'/4 derniers mots de ce tableau contiennent les valeurs $I_{N'/4}(n)$ calculées par l'expression (91). Pour trier le $j^{ième}$ quart du tableau T_R (le raisonnement est identique pour T_I), il suffit de prendre dans T_R une valeur sur quatre à partir de celle qui se trouve en position $I_4(j)$. Ces valeurs sont alors stockées dans l'ordre où on les rencontre dans les N'/4 premières valeurs du tableau S. On commence par j = 1, et après les avoir triées, les N'/4 valeurs ainsi sélectionnées constitueront l'ensemble S_1 qui devra se trouverdans le premier quart du tableau T_R. Pour qu'on puisse les y ranger, on "tasse" donc les 3N'/4 valeurs restantes dans le bas du tableau T_R. Une fois la place nécessaire libérée, on prend la valeur qui a été mise dans la mémoire S(n) et on la stocke dans la mémoire $T_R(m)$, m étant la valeur $I_{N'/4}(n)$ stockée en S(N'/4 + n). Le premier quart du tableau T_R est donc ainsi trié. On recommence pour sélectionner et trier le deuxième quart en prenant une valeur sur trois dans les 3N'/4 valeurs qui avaient été "tassées" dans le bas du tableau T_R et en les stockant dans la première moitié du tableau S..... etc.

En faisant huit opérations de tri successives, on parvient ainsi à ordonner l'ensemble des 2N' valeurs stockées dans T_R et T_I.

II.2.1. β Deuxième partie - Le calcul des intercombinaisons

Le calcul des intercombinaisons est basé sur la remarque suivante : Pour passer d'une étape à l'autre, il est toujours possible de regrouper les valeurs deux par deux de façon à ce que, une fois combinées, elles définissent deux nouvelles valeurs qu'on peut stocker à l'emplacement des anciennes.

Prenons par exemple le cas de la figure 4. Les valeurs C_1 et D_1 peuvent être combinées pour calculer B_1 et B_5 qui seront stockées à leur place.



C'est cette propriété de l'algorithme qui permet de se contenter de 2N' mots en mémoire centrale. Les valeurs initiales $I_0, I_1, \ldots, I_{N'-1}$ de départ sont progressivement modifiées au fur et à mesure des diverses intercombinaisons jusqu'à aboutir aux valeurs $B_0\ B_1, \ldots B_{N'-1}$ du spectre.

Rappelons que le nombre d'étapes d'intercombinaisons à effectuer est égal à $\log_2 N'$.

Une fois le tri terminé, le tableau auxiliaire S sert à stocker les valeurs des cosinus et de sinus permettant le calcul des $W_{N'}^{j}$.

La version "V_3" de ce sous-programme "FFT32T" dont la liste est donnée page 111, a été développée en vue de son utilisation sur un ordinateur disposant d'une mémoire centrale importante et non hiérarchisée. L'accent a donc été mis sur le gain de temps de calcul aux dépens de l'encombrement mémoire. Dans ce but, on remarquera entre autres les points suivants :

- Les boucles "DØ" contenant peu d'opérations ont été développées de façon à diminuer la durée relative des tests correspondant à l'incrémentation de l'indice. C'est le cas en particulier des boucles permettant le tri initial, où il n'y a souvent que des transferts de valeurs.

• De nombreuses variables ont été affectées à des expressions arithmétiques à valeur constante pour éviter d'avoir à les recalculer en différents endroits du programme.

• Toutes les valeurs $W_{N'}^{j}$, sont calculées à partir des N'/4 premières valeurs de cos $\frac{2\pi k}{N'}$ stockées dans le tableau S. De plus pour calculer ces N'/4 valeurs, il n'est fait appel à la fonction sinus (ou cosinus) qu'une fois sur quatre, les trois autres valeurs étant calculées par une formule de récurrence du type :

$$\cos(k\theta + p\theta) = \cos k\theta \cos p\theta - \sin k\theta \sin p\theta$$

avec p = 1,2,3 et k = 1,4,8,...,N'/4

Les valeurs cos pθ et sin pθ sont calculées une fois pour toutes.

• Dans le cas d'un ordinateur à mémoires hiérarchisées, (par exemple quand il existe une antémémoire à cycle de base plus court que pour le reste de la mémoire), il peut être rentable de regrouper dans des zônes de mémoires voisines les valeurs qui doivent être combinées ensemble. Dans ce cas, par exemple, il serait plus avantageux de regrouper la partie réelle et la partie imaginaire d'une valeur complexe dans deux mémoires consécutives au lieu de les mettre dans des tableaux différents comme c'est le cas dans ce sous-programme. Elles auraient ainsi plus de chance de se trouver ensemble dans l'antémémoire au moment de leur utilisation. Des contraintes de programmation externes à ce sous-programme nous ont conduits à garder deux tableaux séparés pour la partie réelle, et la partie imaginaire.

Il existe deux autres versions de ce sous-programme qui sont moins performantes mais qui prennent un peu moins de place en mémoire centrale.

- La version V1 est caractérisée par le fait qu'elle utilise la méthode d'inversion de bits pour faire le tri initial.

- La version V2 est analogue à la version V3, mais elle a 180 instructions de moins.

- Les temps de calcul de la version V3 pour différentes valeurs de N' sont récapitulées sur la figure 7.

- Un mode d'emploi succinct se trouve dans les cartes commentaires au début du sous-programme.

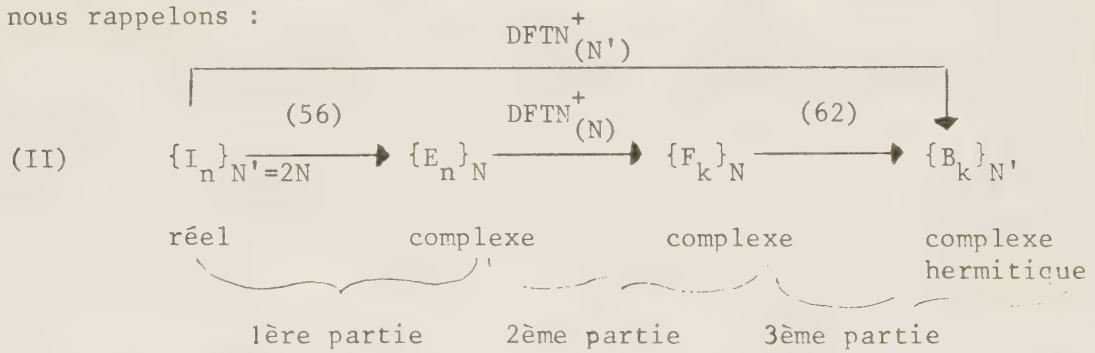
NOTATIONS

Celles du programme se trouvent à droite.

N'	N	T_R	TR
N	NS2	T_I	TI
N'/4	NS4	$2\pi/N'$	PIC
S	S		

II.2.2 - Transformation d'une fonction réelle en mémoire centrale - Programme "FFT64K"

Nous avons vu en II.1.2 que le calcul de la DFTN d'une fonction réelle peut se faire à l'aide de l'algorithme résumé dans le schéma (II) que nous rappelons :



A l'entrée du sous-programme, pour stocker $\{I_n\}_{N'}$, il faut disposer de N' mots en mémoire centrale. D'autre part, en sortie, les N' valeurs complexes B_k sont réparties sur un domaine $2\Delta\nu'$ (fig. 2,b) égal au double de l'intervalle spectral libre 1/2 h. De plus B(ν) étant hermitique, il est suffisant de connaître les N=N'/2 valeurs complexes qui correspondent à l'intervalle spectral libre où se trouve le spectre cherché. Suivant la valeur du paramètre "IIR" du sous-programme on ne calculera que B_k ou B_{k+N} pour k = (0,1,...,N-1). Pour la sortie, il faut donc aussi disposer de N' mots en mémoire centrale, mais étant donné qu'il s'agit de N valeurs complexes, on a conservé la disposition des deux tableaux T_R et T_I de N mots réservés l'un

pour la partie réelle et l'autre pour la partie imaginaire.

Comme nous le voyons sur le schéma d'ensemble (II), l'algorithme que nous étudions peut se diviser en trois parties principales.

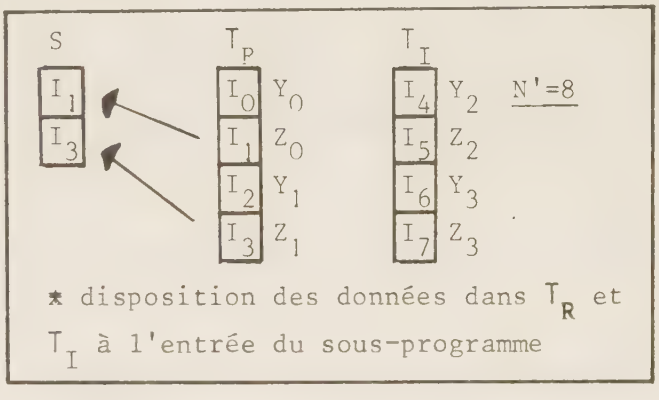
II.2.2. α Première partie - Constitution de $\{E_n\}_N$ à partir de $\{I_n\}_{N'}$

Les relations (56) permettant de passer de $\{I_n\}_{N'}$ à $\{E_n\}_N$ sont :

$$E_n = Y_n + i\, Z_n \quad \text{pour } n = 0,1,2,\ldots,N-1$$

avec $\begin{cases} \{Y_n\}_N = \{I_0, I_2, I_4, \ldots, I_{2N-2}\} \\ \{Z_n\}_N = \{I_1, I_3, I_5, \ldots, I_{2N-1}\} \end{cases}$

La présence d'un tableau auxiliaire S de N'/4 mots en mémoire centrale facilite bien la sélection et le réarrangement des valeurs Y_n et Z_n.



* disposition des données dans T_R et T_I à l'entrée du sous-programme

Comme le montre le schéma ci-contre, les N premières valeurs de $\{I_n\}_{N'}$ doivent se trouver dans T_R et les N dernières dans T_I au moment de l'appel du sous-programme "FFT64K". On prend donc une valeur sur deux dans T_R à partir de I_1 et on stocke les N/2 valeurs Z_n (n=0, 1,...,N/2-1) ainsi sélectionnées dans le tableau S. En même temps, on range par "tassement progressif" les valeurs restantes Y_n (avec n=0,1,...,N/2-1) dans la première moitié de T_R. On peut alors stocker dans la deuxième moitié de T_R laissée libre les valeurs Y_n (avec n=N/2,N/2+1,...,N-1) contenues dans T_I. Il ne reste plus alors qu'à ranger dans la deuxième moitié de T_I les valeurs Z_n (avec n=N/2,N/2+1,...,N-1) puis à transférer le tableau S dans la première moitié de T_I.

II.2.2. β Deuxième partie - Le calcul de la $DFTN^{+}_{(N)}$

Une fois la première partie du programme effectuée, la partie réelle de $\{E_n\}_N$ se trouve rangée dans les N mots de T_R et la partie imaginaire dans T_I. On dispose d'autre part du tableau auxiliaire de N/2 mots. Nous nous trouvons donc dans les conditions d'application du sous-programme "FFT32T" qui calcule la $DFTN^{+}_{(N)}$ d'une fonction complexe. La deuxième partie du sous programme "FFT64K" est donc une reproduction du sous-programme "FFT32T".

A la sortie de cette deuxième partie on trouve donc dans T_R et T_I le partie réelle et la partie imaginaire de $\{F_k\}_N$ qui est la $DFTN^{+}_{(N)}$ de $\{E_n\}_N$.

II.2.2. ϒ Troisième partie - Le calcul de la fonction hermitique

A partir de $\{F_k\}_N$, il s'agit de calculer $\{B_k\}_N$ ou $\{B_{N+k}\}_N$ par l'expression (62) ci-dessous :

$$B_k = \frac{F_k + F^*_{N-k}}{2} + W^k_{N'} \frac{F_k - F^*_{N-k}}{2i}$$

$$B_{N+k} = \frac{F_k + F^*_{N-k}}{2} - W^k_{N'} \frac{F_k - F^*_{N-k}}{2i}$$

L'algorithme correspondant est facile à réaliser si on remarque que :

$$B_{N-k} = \frac{F_{N-k} + F^*_k}{2} + W^{-k}_{N'} \frac{F_{N-k} - F^*_k}{2i}$$

Autrement dit que : $\begin{cases} B_k = f\ (F_k, F_{N-k}) \\ B_{N-k} = f'\ (F_k, F_{N-k}) \end{cases}$

à partir des deux valeurs F_k et F_{N-k} on définit par (62) deux valeurs B_k et B_{N-k} qui seront stockées à leur place. Aucun tableau auxiliaire n'est nécessaire dans cette troisième partie.

REMARQUES GENERALES :

- C'est la version V3 de ce sous-programme qui est listée page 113. Elle a été développée dans le même esprit que la version V3 du sous-programme

"FFT32T" et toutes les remarques concernant les techniques de programmation tendant à diminuer le temps d'exécution, sont valables pour cette version.

- Pour ce sous-programme aussi, il existe deux autres versions V1 et V2 de taille mémoire plus réduite, mais moins performantes.

- Les temps de calcul de la version V3 sont récapitulés sur la figure 7.

- Un mode d'emploi succinct se trouve dans les cartes commentaires au début du sous-programme.

NOTATIONS PRINCIPALES

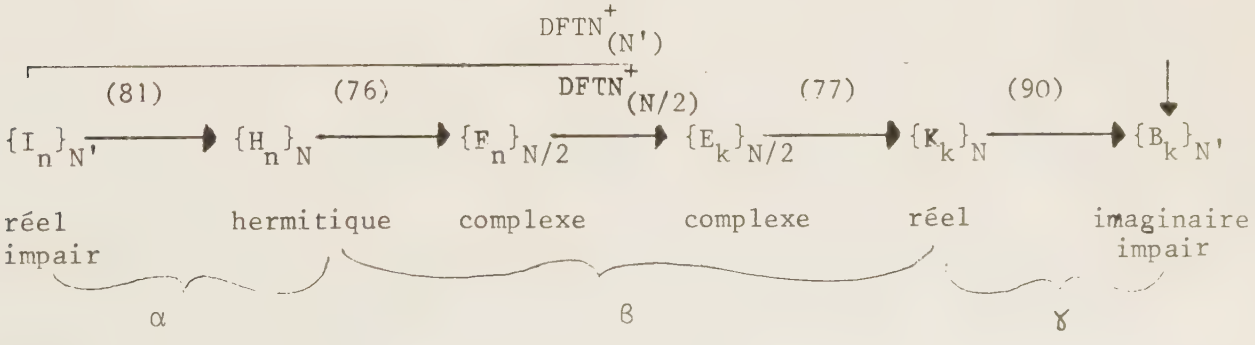
(celles du programme sont à droite)

T_R	TR	N	N
T_I	TI	NS2	N/2
S	S	N'	ND

II.2.3. - Transformation d'une fonction réelle impaire en mémoire centrale - Programme "FFT128"

Reprenons le schéma (IV) qui définissait en II.1.3 l'algorithme adapté au calcul de la DFTN d'une fonction impaire.

(IV)



Rappelons que I(t) est une fonction échantillonnée réelle, impaire, et périodique de période N'=2N du type de celle représentée sur la figure 2,d.

$\{I_n\}_{N'}$ est l'ensemble des N' échantillons qui partent de l'origine et nous avons vu en (78) qu'il était possible de remplacer les N derniers échantillons de cet ensemble par des zéros pour ne conserver que $\{I_n\}_N$.

En entrée, le sous-programme "FFT128" n'aura donc besoin que de N mots en mémoire centrale pour stocker $\{I_n\}_N$.

En sortie, $\{B_k\}_{N'}$ est un ensemble de N' valeurs de la fonction réelle $B_i(\nu)$ (figure 2,d) qui est impaire et périodique. Les N' valeurs de cet ensemble sont réparties sur une période qui recouvre sur une de ses deux moitiés l'intervalle spectral libre $\Delta\nu'$ où se trouve le spectre cherché (triangle hachuré sur la figure) et sur l'autre moitié, son image antisymétrique. Pour la sortie aussi, on se contentera donc de N mots en mémoire puisqu'il est suffisant de calculer les N valeurs réelles de $B(\nu)$ qui correspondent à l'intervalle spectral libre désiré. Ce choix se fait en fonction du paramètre "IR" dans le sous-programme.

Si ce paramètre vaut 1 c'est la première moitié de la période de $B_i(\nu)$ qui est calculée et s'il est supérieur à 1 c'est la deuxième moitié.

Ces N mots en mémoire centrale sont réservés sous la forme de deux tableaux T_R et T_I de N/2 mots pour pouvoir utiliser le sous-programme "FFT32T" vu plus haut pour calculer la DFTN de $\{F_n\}_{N/2}$. Dans le même but, nous réservons un tableau auxiliaire S de N/4 mots.

L'ensemble de ces trois tableaux T_R, T_I, S est suffisant pour programmer l'ensemble de l'algorithme défini par le schéma (IV). Cet algorithme se divise en trois parties principales α, β, γ.

II.2.3. α Constitution de la fonction hermitique H

Comme dans le cas de la fonction réelle, la première moitié de $\{I_n\}_N$ est stockée dans le tableau T_R et la deuxième moitié dans T_I.

Les valeurs H_n de la fonction hermitique H seront calculées par l'expression (81) :

$$H_n = \frac{1}{2}\left[I_{2n+1} - I_{2n-1}\right] + i\, I_{2n}$$

et comme $H^{*}_{\frac{N}{2}+p} = H_{\frac{N}{2}-p}$

on se contente de calculer H_n pour n = (0,1,2,...,N/2 - 1) soit $\{H_n\}_{N/2}$ et on commence par la partie réelle. Pour ce faire, on remplace progressivement les valeurs initiales I_n du tableau T_R par la valeur réelle de H_n qui vient d'être calculée et on sauvegarde dans le tableau S les valeurs $(I_2, I_4,...,I_{N/2})$ qui ne sont pas utilisées dans la partie réelle.

La partie imaginaire de $\{H_n\}_{N/2}$ soit : $\{I_0, I_2,...,I_{N-2}\}$ sera stockée dans T_I. Une fois le calcul des parties réelles terminé, on range donc dans le bas de T_I les valeurs $(I_{N/2}, I_{N/2+2},..., I_{N-2})$ puis on transfère dans la première moitié de T_I ainsi libérée les valeurs qui avaient été sauvegardées dans S.

II.2.3. β Calcul de la DFTN de la fonction hermitique H.

A partir des valeurs de $\{H_n\}_{N/2}$ on calcule les valeurs complexes F_n en utilisant la formule (76) ce qui donne :

$$(92) \qquad F_n = \frac{1}{2}\left[H_n + H^{*}_{\frac{N}{2}-n} + i\, W_N^n \left(H_n - H^{*}_{\frac{N}{2}-n}\right)\right]$$

pour n = 0,1,2,..,N/2-1

Cette formule est tout à fait semblable à la formule (62) dont la programmation a été vue en II.2.2.δ. On peut donc se reporter à ce paragraphe pour cette partie de l'algorithme.

Cette deuxième partie β de l'algorithme est repérée par le commentaire "SUBROUTINE FFT65K" dans la version V3 du sous-programme "FFT128" listée page 116.

Lorsque l'ensemble $\{F_n\}_{N/2}$ a été calculé d'après (92), il reste à en calculer la $DFTN^{+}_{(N/2)}$ par le programme "FFT32T" qui a été reproduit à cet endroit là. Pour cette partie de l'algorithme on peut donc se reporter au paragraphe II.2.1. On obtient alors l'ensemble $\{E_k\}_{N/2}$ de N/2 valeurs complexes à partir desquelles on définit par la relation (77) l'ensemble $\{K_k\}_N$ de N valeurs réelles.

Nous avons en effet :

(93) $$E_k = K_{2k} + i\, K_{2k+1} \qquad (k = 0,1,2,..,N/2-1)$$

La partie réelle de $\{E_k\}_{N/2}$ se trouve dans T_R et la partie imaginaire dans T_I.

La première moitié du tableau T_I est transférée dans S. Il s'agit en fait des valeurs $K_1, K_3, \ldots, K_{N/2-1}$. Les autres valeurs de T_I peuvent alors être rangées à leur emplacement final dans ce même tableau. Les N/4 dernières valeurs du tableau T_R c'est-à-dire les valeurs $K_{N/2}, K_{N/2+2}$ viennent ensuite compléter le tableau T_I.

Les N/4 premières valeurs de T_R sont alors rangées dans ce même tableau à leur place finale et les valeurs qu'on avait stockées en S viennent s'intercaler entre-elles pour le compléter.

II.2.3. γ Le calcul final des N valeurs B_k.

A la fin du tri précédent, on trouve dans T_R les N/2 premières valeurs de $\{K_k\}_N$ et dans T_I les N/2 dernières.

La relation (90) permet alors de calculer $\{B_k\}_N$, à partir de $\{K_k\}_N$. A une constante imaginaire près, elle s'exprime par :

(94) $$B_k = \frac{K_k + K_{N-k}}{\sin \frac{\pi k}{N}} - K_k + K_{N-k}$$

La programmation de cette expression est facilitée si on remarque que :

(95) $$B_{N-k} = \frac{K_{N-k} + K_k}{\sin \frac{\pi k}{N}} - K_{N-k} + K_k$$

On voit en effet, que le couple de valeurs (B_k, B_{N-k}) s'exprime uniquement en fonction du couple (K_k, K_{N-k}) qu'il peut remplacer après avoir été calculé.

D'autre part, nous avons rappelé au début de ce paragraphe que les valeurs de $\{B_k\}_N$, étaient réparties de façon impaire par rapport à la valeur B_N se trouvant au centre d'une période de $B_i(\nu)$. On a donc :

$B_{N+k} = -B_{N-k}$ d'où d'après (95) :

(96)
$$B_{N+k} = \frac{K_k + K_{N-k}}{-\sin \frac{\pi k}{N}} - K_k + K_{N-k}$$

Par comparaison avec (94), on voit qu'il suffira par exemple de changer le signe du sinus pour sélectionner les N valeurs B_k se trouvant dans l'intervalle spectral libre désiré.

Remarque concernant la version V3 du sous-programme "FFT128"

La réalisation de la troisième partie ϒ de cet algorithme, nécessite la connaissance des valeurs de $\sin \frac{\pi k}{N}$. Ces valeurs ont bien été déjà calculées et stockées dans le tableau auxiliaire S au moment du calcul de la $DFTN^+_{(N/2)}$ de $\{F_n\}_{N/2}$ mais elles ont été détruites au moment du tri qui permet de passer de $\{E_k\}_{N/2}$ à $\{K_k\}_N$. Il faut donc recalculer ces valeurs ce qui est le cas dans les versions V1 et V2 de ce sous-programme.

Pour gagner du temps dans la version V3, nous avons inversé le tri des valeurs E_k et la partie ϒ de l'algorithme, ce qui évite d'avoir à recalculer les valeurs de $\sin \frac{\pi k}{N}$. C'est cette version V3 qui est listée page 116. Elle a été développée dans le même esprit que la version V3 des deux sous-programmes déjà vus. Les remarques faites en II.2.1.β qui concernent les techniques de programmation tendant à diminuer le temps d'exécution sont toujours valables. Les temps de calcul pour différentes valeurs de N' sont récapitulés sur la figure 7 et un mode d'emploi succinct est donné dans les cartes commentaires au début du programme.

NOTATIONS PRINCIPALES (Celles du programme sont à droite)

T_R	TR	N/2	N
T_I	TI	N/4	NS2
S	S		

II.2.4. - Transformation d'une fonction réelle impaire nécessitant l'utilisation d'un support auxiliaire de mémoire sur disque magnétique - Programme "TFID4"

Nous avons vu en II.2.3 que pour calculer la DFTN d'une fonction réelle impaire de N échantillons, le sous-programme "FFT128" avait besoin de $(N + N/4 + N_p)$ mots en mémoire centrale, la constante N_p étant égale au nombre de mots occupés par le programme proprement dit. Si N_m est le nombre de mots libres dans la mémoire centrale de l'ordinateur utilisé, le plus grand nombre $N=2^n$ d'échantillons qui peut être transformé sera alors fixé par la relation :

(97) $$N_m > N + N/4 + N_p$$

Pour donner un ordre de grandeur, avec une mémoire de 256 K mots (K = 1024), la limite supérieure pour N est 128 K.

Pour s'affranchir de cette limitation, il faut donc disposer d'un programme capable de pallier le manque de place en mémoire centrale par l'utilisation de mémoires auxiliaires sur un organe périphérique de l'ordinateur. Nous avons donc écrit un programme "TFID4" utilisant des blocs de données stockés sur disque magnétique en accés direct. Ce programme est caractérisé par une grande souplesse d'adaptation à divers types d'installation de calcul de façon à pouvoir résoudre les deux situations α et β suivantes :

- α - On dispose d'un gros ordinateur avec beaucoup de mémoires (512K mots par ex.) mais le nombre N d'échantillons à transformer est très grand. C'est le cas par exemple des interférogrammes que nous devons transformer puisque le plus souvent leur nombre d'échantillons est compris entre 10^6 et 2.10^6. Dans ce cas il faudrait alors disposer en mémoire centrale de 2600K mots environ pour utiliser le sous-programme "FFT128".

- β - Le nombre N de valeurs à transformer est de l'ordre de quelques dizaines de mille, mais on ne dispose que d'un petit ordinateur auquel on peut connecter une unité de disque magnétique, mais dont la taille mémoire réduite est incompatible avec l'utilisation du sous-programme "FFT128". C'est le cas par exemple lorsqu'il faut transformer 80 000 valeurs et qu'on ne dispose que de 32K mots en mémoire centrale.

Nous avions écrit en 1969 un premier programme appelé "FFT1024K" [36] utilisant des ensembles de blocs de mots (ou fichiers) sur disque magnétique. Il nous a permis de traiter sur le 360/75 I.B.M. (256K mots) du CIRCE les premiers interférogrammes de 10^6 valeurs. Pour N=1024K, le temps de calcul était de 22 minutes. Ce programme avait été écrit à partir de l'algorithme de transformation d'une fonction complexe (défini en II.1.1) en l'adaptant au cas d'une fonction réelle impaire, mais même avec ces adaptations cet algorithme restait moins performant sur le plan encombrement mémoire et durée de calcul que celui défini en (II.1.3) pour le calcul de la DFTN d'une fonction réelle impaire. Néanmoins, avec quelques modifications il peut être utile pour transformer des fonctions réelles sans caractère de symétrie. Sous cette forme modifiée, et avec le nom de "TFRD", il a d'ailleurs été utilisé par H.P. Larson au "Lunar and Planetary Laboratory" de Tucson pour transformer des interférogrammes réels de l'ordre de 60 000 valeurs sur un 1130 I.B.M. en n'utilisant que 8192 valeurs en mémoire centrale.

Nous avons donc écrit le programme "TFID4" en partant de l'algorithme défini en II.1.3 et représenté sur le schéma IV. La programmation de cet algorithme a déjà été étudiée au paragraphe II.2.3 précédent et sur le plan des principes tout ce qui en a été dit reste valable pour ce programme. Nous allons donc nous contenter de développer les particularités propres à l'utilisation des fichiers sur disque magnétique.

Nous commencerons par donner le principe de base permettant de remplacer la $DFTN_{(N)}$ de N valeurs par n_b $DFTN_{(N_T)}$ de N_T valeurs avec $N_T=N/n_b$ puis nous montrerons comment toutes les étapes de l'algorithme du schéma IV peuvent être réalisées à partir de petits blocs de données lus sélectivement sur disque puis transférés dans la mémoire centrale.

II.2.4. α Principe de base.

Nous avons vu en II.1.1 à propos de la technique de "decimation in time" que la $DFTN_{(N)}$ de n'importe quelle fonction complexe est une combinaison linéaire de la $DFTN_{(N/2)}$ de deux fonctions ayant moitié moins de points et issues de la première. C'est ce qu'illustrait la figure 4.

Si on applique p fois ce principe, on voit qu'il est toujours possible de remplacer une $DFTN_{(N)}$ par n_b $DFTN_{(N_T)}$ ne portant que sur N_T valeurs avec :

$$n_b = 2^p \quad \text{et} \quad N_T = N/n_b$$

Ces n_b $DFTN_{(N_T)}$ seront suivies de p étapes d'intercombinaisons. La figure 8 en donne une illustration dans le cas où N=16, n_b=4, N_T=4.

Les n_b $DFTN_{(N_T)}$ doivent être précédées d'une étape de tri permettant de définir les n_b séries de N_T valeurs auxquelles il faut les appliquer. Dans le cas de la figure 8, on voit que les quatre séries D_1, D_2, D_3, D_4 sont identiques à celles de la figure 6. Nous allons donc utiliser les remarques faites au paragraphe II.2.1.α pour définir ces séries.

Pour sélectionner les N_T valeurs de la $j^{ième}$ série D_j, il suffit de prendre dans la série initiale de N valeurs, une valeur sur n_b en partant de celle qui se trouve en position $I_{n_b}(j)$. (Voir (91)).

Dans le programme "TFID4", ce principe est appliqué au calcul de la DFTN des N/2 valeurs complexes F_n (voir schéma IV en II.2.3), qui sera ainsi remplacé successivement par :

- le tri et la sélection des n_b séries de N_T valeurs.
- le calcul de la $DFTN_{(N_T)}$ de chaque série.
- le calcul des p étapes d'intercombinaisons.

Comme la DFTN initiale ne portait que sur N/2 valeurs, nous aurons :

$$(98) \quad \begin{cases} n_b = 2^p \\ N_T = (N/2)/n_b \end{cases}$$

Le calcul des $DFTN_{(N_T)}$ (voir II.2.1) nécessite la réservation en mémoire centrale de deux tableaux T_R et T_I de N_T mots et d'un tableau auxiliaire S de $N_{T/2}$ mots. Pour réaliser les autres étapes du programme telles que le tri ou les intercombinaisons, nous devons réserver trois fichiers de N/2 mots sur disque magnétique. Ces fichiers seront repérés par leur numéro <1>, <11>

et <21>. Les N/2 mots y sont groupés par blocs de l valeurs. Il faut donc que l soit un sous multiple de N/2 et l'unité de disque utilisée doit permettre un accès direct à chacun de ces blocs repérés par leur numéro à l'intérieur de chaque fichier.

L'adaptation du programme "TFID4", d'une part aux caractéristiques de l'installation utilisée et d'autre part au nombre de valeurs des fonctions à transformer se fait en adaptant la valeur des paramètres : N_T et l. En particulier, N_T sera choisi en fonction de la place mémoire disponible. Si N_m est le nombre de mots disponibles en mémoire centrale et si N_p est le nombre de mots occupés par les instructions du programme et tout ce qui est nécessaire à son exécution, nous devons satisfaire la relation suivante :

$$N_m \geq 2.\ N_T + N_T/2 + N_p \qquad (99)$$

La valeur de N_T ayant été choisie et fixée au moment de l'adaptation du programme (par la modification des ordres "DIMENSION" et "EQUIVALENCE"), la valeur du paramètre n_b sera calculée automatiquement par la relation (98) en fonction de N qui peut varier d'un calcul à l'autre.

Le choix de l, le nombre de valeurs par bloc sur les trois fichiers réservés sur disque, est lié aux caractéristiques techniques de l'unité de disque utilisée et à la valeur de N_T choisie. Sans entrer dans les détails, disons qu'il faut prendre pour l une puissance de 2 qui soit un sous multiple de N_T et qui permette :

- d'obtenir un bon coefficient de remplissage du disque.
- de minimiser le nombre de transferts entre le disque et la mémoire.
- de stocker simultanément en mémoire dans les tableaux T_R et T_I plusieurs de ces blocs de l valeurs. (Au moins huit comme nous le verrons).

Tous les détails pratiques concernant le choix de ces deux paramètres N_T et l sont d'ailleurs donnés dans les cartes commentaires du début du programme qui est listé

REMARQUE :

1 correspond au plus petit nombre de valeurs qu'on peut lire ou écrire sur le disque. L'écriture ou la lecture d'un bloc de kl valeurs est toujours possible si k est un entier supérieur ou égal à 1.

II.2.4 β Acquisition des données et constitution de la fonction hermitique H

Nous avons vu en II.2.3.α que les valeurs H_n de la fonction hermitique sont calculées par :

$$\begin{cases} H_n = \frac{1}{2}\left[I_{2n+1} - I_{2n-1}\right] + i\, I_{2n} \\ \text{pour } n=0,1,2,\ldots,N/2-1 \end{cases}$$

Comme on le voit le calcul peut se faire séquentiellement puisque pour chaque valeur H_n il suffit de connaître trois valeurs consécutives de $\{I_n\}_N$. Au moment d'exécuter le programme, les valeurs de l'ensemble $\{I_n\}_N$ doivent se trouver sur une bande magnétique par blocs de p_1 valeurs. Cette valeur de p_1 est à fixer au moment de l'adaptation du programme. Le nombre de valeurs à lire sur la bande est m. C'est un multiple de p_1 inférieur ou égal à N.

(100) $\quad m = K\, p_1 \leq N$

Le programme lit donc un bloc de p_1 valeurs I_n sur la bande et le transfère en mémoire dans un tableau D prévu à cet effet. Ces valeurs sont alors traitées séquentiellement pour définir les valeurs H_n dont la partie réelle est progressivement stockée dans T_R et la partie imaginaire dans T_I. Lorsque ces deux derniers tableaux sont pleins, ils sont transférés respectivement sur les fichiers <I> et <II>. Lorsque les p_1 valeurs de D ont été utilisées, le programme lit un nouveau bloc sur la bande, et ce jusqu'au traitement complet des m valeurs I_n à lire.

- La version du programme "TFID4" listée page 120,

a été modifiée pour pouvoir être essayée en simulation. Dans ce but on a

remplacé la lecture de la bande magnétique par le calcul de blocs de p_1 valeurs I_n. On a aussi choisi des valeurs très petites pour N_T, p_1 et 1. Enfin a titre de vérification on a ajouté à la fin du programme le tracé sur imprimante d'une partie du spectre calculé. Ce tracé sur imprimante est réalisé par le sous-programme "COURB" et toutes les instructions ajoutées pour permettre la simulation sont repérées par l'inscription "SIMU" en colonnes 73 à 80.

- Les p_1 valeurs du tableau D réservé en mémoire centrale pour la lecture de la bande magnétique font partie du paramètre N_p dans l'expression (97).

II.2.4 ɣ Constitution de $\{F_n\}_{N/2}$

Nous avons déjà vu en II.2.3.β le calcul des valeurs F_n à partir des valeurs H_n. D'après (92), nous avons :

$$(101) \quad \begin{cases} F_n = f(H_n, H_{N/2-n}) \\ F_{N/2-n} = f'(H_n, H_{N/2-n}) \end{cases}$$

Il est donc avantageux de faire un traitement simultané de blocs de valeurs réparties symétriquement par rapport à $H_{N/4}$ et $F_{N/4}$. Les parties réelles et imaginaires de H_n et $H_{N/2-n}$ sont lues par blocs de 1 valeurs sur disque. Il faut donc prévoir 4 tableaux en mémoire centrale pour les y stocker. Il faut aussi prévoir 4 tableaux pour conserver les valeurs complexes F_n et $F_{N/2-n}$ au fur et à mesure qu'elles sont calculées. Une fois pleins, ces quatre derniers tableaux seront transférés sur disque. Nous avons donc réservé huit tableaux de $N_T/4$ mots qui sont en équivalence sur le $2.N_T$ mots des tableaux T_R et T_I déjà vus. Les quatre blocs correspondant aux parties réelles et imaginaires des 1 valeurs H_n et $H_{N/2-n}$ lues dans <I> et <II> sont stockées dans les tableaux T_{r1}, T_{r2}, T_{i1}, T_{i2} et les quatre blocs correspondant aux valeurs F_n et $F_{N/2-n}$ dans les tableaux B_{r1}, B_{r2}, B_{i1}, B_{i2}. Ces huit tableaux sont caractérisés par deux indices. Le premier indice indique si le tableau contient des parties réelles ((r)) ou des parties imaginaires ((i)) et le deuxième indice indique si le tableau contient des valeurs d'indice n ((1)) ou des valeurs d'indice N/2-n ((2)). A la fin de cette partie du calcul, les parties réelles des valeurs F_n se trouvent en <I> et les parties imaginaires en <II>.

II.2.4 δ Sélection et tri des n_b séries dont il faut calculer la DFTN.

Nous avons vu en II.2.4.α comment il fallait constituer les n_b séries D_j de N_T valeurs F_n. Nous avons d'autre part d'après (98) :

(102) $$n_b = (N/2)/N_T$$

Nous allons envisager le cas des valeurs réelles, mais le même traitement sera appliqué aux valeurs imaginaires. Le tri peut se faire de façon séquentielle en lisant progressivement les valeurs de la partie réelle de F_n. On lit donc un bloc de $N_T/2$ valeurs réelles de F_n sur disque et on le transfère en mémoire centrale dans le tableau S de $N_T/2$ mots. Ces valeurs sont triées et stockées dans un autre tableau T_{RI} de $2N_T$ mots mis en équivalence sur T_R et T_I. Ce tableau T_{RI} sera complétement rempli lorsqu'on aura transféré 4 blocs par l'intermédiaire du tableau S. On se trouve alors en présence de $2\ N_T$ valeurs qu'il faut répartir dans les n_b séries D_j. Pour chacune de ces séries, on n'a calculé que n_8 valeurs avec :

(103) $$n_8 = 2.\ N_T/\ n_b$$

Pour libérer le tableau T_{RI}, il faut pouvoir transférer les n_8 valeurs de chaque série D_j à l'emplacement prévu pour elle sur le disque. Ceci n'est techniquement possible que si :

(104) $$\boxed{n_8 = k\ 1 \quad \text{avec } k \geqslant 1}$$

Si cette condition n'est pas réalisée, il n'est pas toujours rentable de diminuer la valeur choisie pour 1, car en le faisant on multiplie le nombre de transferts disque$\longleftrightarrow$mémoire centrale ce qui pénalise au point de vue performance toutes les autres étapes de ce programme.

Nous avons donc prévu la possibilité d'utiliser un quatrième fichier <22> sur disque. Ses blocs sont plus petits et égaux à 1_{22} qui doit prendre la plus petite valeur prévisible de n_8. Dans ce fichier <22>, il faut réserver n_{22} blocs de 1_{22} valeurs avec :

(105) $$\begin{cases} 1_{22} = \text{Inf.}\left[n_8 \right] & \text{(Inf. = val. infér. de)} \\ n_{22} = (1/1_{22}).n_b \end{cases}$$

Il est alors possible de libérer T_{RI} en vidant ses 2 N_T valeurs par blocs de n_8 valeurs dans le fichier <22> . Ce n'est que lorsque 1 valeurs auront été ainsi accumulées dans chaque série qu'il sera alors possible de les transférer du fichier <22> vers le fichier <11> normal. A la fin du tri, les valeurs réelles des séries D_j sont stockées sur <11> et les valeurs imaginaires sur<21>.

Le tableau ci-dessous montre pour une configuration donnée du programme "TFID4" les variations relatives de N, n_b, n_8. Dès que n_8 devient inférieur à 1, l'utilisation de fichier <22> devient nécessaire, et le tableau indique alors le nombre n_{22} de blocs qu'il faut réserver pour ce fichier.

	N_T = 8K, l = 1024			rappel (k = 1024)		
N	64K	128K	256K	512K	1024K	2048K
n_b	4	8	16	32	64	128
$n_8 (= l_{22})$	4K	2K	1024K	512	256	128
n_{22}	0	0	0	64	256	1024
				fichier <22> nécessaire		

Dans la configuration utilisée au CIRCE, nous avons pris les valeurs l_{22} et n_{22} correspondant à la dernière colonne. Ce programme permet alors le calcul des interférogrammes jusqu'à 2.10^6 valeurs et il n'utilise le fichier <22> que si N > 256K.

La DFTN de chacune des séries D_j de N_T valeurs ainsi triées est faite par le sous-programme "FFT32T" version "V31" qui ne diffère de la version "V3" étudiée en II.2.1 que par le fait qu'il a besoin d'un tableau supplémentaire "W" de $N_T/4$ mots pour garder les valeurs des sinus nécessaires au calcul de la DFTN. La première fois qu'on l'appelle l'argument "IWW" vaut 1 et il calcule ce tableau de sinus. Les (n_b-1) autres appels à ce sous-programme se font en donnant la valeur zéro à l'argument "IWW" ce qui lui indique qu'il peut directement utiliser le tableau des sinus qu'il a créé la première fois.

Le calcul de la DFTN des $n_b = 2^p$ séries D_j est suivi de p étapes d'intercombinaisons (fig. 8) dont la programmation a déjà été étudiée en II.2.1.β. En s'y reportant, on verra que là aussi il y a intérêt à regrouper en mémoire centrale deux blocs (non contigüs sur le disque) de valeurs

complexes pour les combiner et définir ainsi deux autres blocs de valeurs complexes qui prendront la place des anciens sur le disque. On est donc amené à utiliser les huit tableaux déjà énumérés plus haut, c'est-à-dire T_{r1}, T_{r2}, T_{i1}, T_{i2} pour les blocs à combiner qui sont lus sur les fichiers <11> et <21>, et B_{r1}, B_{r2}, B_{i1}, B_{i2} pour le résultat des intercombinaisons. La signification des deux indices est la même que celle définie à la fin de II.2.4.γ.

II.2.4. ε Calcul du spectre $\{B_k\}_N$, à partir de $\{E_k\}_{N/2}$

Dans une première étape, il faut passer des N/2 valeurs complexes E_k stockées en <11> et <21> aux N valeurs réelles K_k. (Voir schéma IV en II.2.3).

$$\{E_k\}_{N/2} \xrightarrow{(77)} \{K_k\}_N$$

complexe réel

D'après (93) nous avons : $E_k = K_{2k} + i\, K_{2k+1}$

$$E_0 \xrightarrow{(93)} (K_0, K_1)$$

$$E_1 \longrightarrow (K_2, K_3)$$

etc...

Cette partie de l'algorithme peut donc se faire de façon séquentielle. A toute valeur complexe E_k lue on fait correspondre deux valeurs réelles K_k.

Nous avons réservé quatre tableaux de $N_T/2$ mots en mémoire centrale. Il s'agit de T_{xr} et T_{xi} qui sont en équivalence sur T_R et de B_{x2} et B_{xi} qui sont en équivalence sur T_I. Pour transférer en mémoire $N_T/2$ valeurs E_k, le bloc correspondant à la partie réelle est lu sur <11> puis stocké en T_{xr} et le bloc correspondant à la partie imaginaire est lu sur <21> puis stocké en T_{xi}. Les N_T valeurs K_k qui sont issues de ces deux blocs, sont rangées dans T_I puis transférées dans <1> pour les N/2 premières valeurs de $\{K_k\}_N$ et dans <11> pour les N/2 dernières valeurs.

Dans la dernière étape de ce programme, (étape) sur le schéma IV) il faut passer des valeurs K_k aux valeurs B_k

$$\{K_k\}_N \xrightarrow{(90)} \begin{cases} \{B_k\}_N & \text{si le paramètre "IR"=1} \\ \{B_{k+N}\}_N & \text{" "IR">1} \end{cases}$$

réel

Nous avions remarqué en II.2.3.γ que le couple de valeurs (B_k, B_{N-k}) s'exprimait uniquement en fonction du couple de valeurs (K_k, K_{N-k}). Comme il s'agit cette fois de valeurs réelles, il suffira de lire deux blocs de $N_T/2$ valeurs K_k répartis symétriquement par rapport à $K_{N/2}$, puis de les combiner suivant la relation (90) pour obtenir deux blocs de $N_T/2$ valeurs B_k qui pourront être transférés à la place des deux premiers sur le disque. Les deux blocs lus sont stockés en mémoire centrale dans T_{xr} et T_{xi}. Les deux blocs calculés sont stockés dans B_{xr} et B_{xi}. En fait, ces deux derniers blocs ne peuvent pas être complétement calculés à partir des deux premiers. Il faut lire les deux blocs de valeurs K_k suivants pour finir de les calculer et pouvoir les transférer sur disque ce qui complique l'algorithme. On tombe d'ailleurs sur le même inconvénient lorsqu'on programme la partie de l'algorithme correspondant au paragraphe II.2.4.γ.

A la fin du programme "TFID4", on trouve donc les N/2 premières valeurs B_k dans le fichier <1> et les N/2 dernières dans le fichier <11>.

REMARQUES GENERALES

- Les remarques faites en II.2.1.β qui concernent les techniques de programmation tendant à diminuer le temps d'exécution sont valables pour la version de ce programme listée

- Les temps de calcul pour différentes valeurs de N' sont récapitulés sur la figure 7.

- L'encombrement en mémoire du programme proprement dit (voir N_p en (97)) peut être largement diminué si on le désire car toutes les étapes que nous avons décrites peuvent être faites séparément les unes après les autres en utilisant les fichiers disques <1>, <11> et <21> comme lien de l'une à l'autre. Notons enfin que comme pour les autres programmes il en existe des versions moins performantes mais ayant moins d'instructions, donc un encombrement en mémoire plus faible. Dans le même but, il est facile de remplacer le sous-programme "FFT32T"-V31" par une version de taille plus réduite (FFT32T-V2 par exemple).

- Un mode d'emploi détaillé est donné sous forme de cartes commentaires au début du programme.

NOTATIONS (Celles du programme sont à droite).

Tableaux

T_{RI}	TRI
T_{R}	TR
T_{I}	TI
D	D
S	S
T_{r1}	TR1
T_{r2}	TR2
T_{i1}	TI1
T_{i2}	TI2
B_{r1}	BR1
B_{r2}	BR2
B_{i1}	BI1
B_{i2}	BI2
T_{xr}	TXR
T_{xi}	TXI
B_{xr}	BXR
B_{xi}	BXI
W	W

Variables

N	N
m	M
N_T	NDIMTR
n_b	INBL
p	IPUI
1	LSIZE
p_1	JPL
n_8	NPM8
1_{22}	MSIZE
n_{22}	NLS

III. - INTERPOLATION DANS LE SPECTRE

III.1.1. - Principe de l'interpolation sans apodisation.

Les interférogrammes que nous devons transformer étant des fonctions impaires, nous allons étudier l'interpolation du spectre d'une fonction de ce type, mais l'ensemble de cette étude peut s'appliquer au cas de la fonction paire. Nous partons donc de deux fonctions $I_i(t)$ et $B_i(\nu)$ (fig. 2 d) étudiées en I.5.1 et II.1.3 qui sont transformées l'une de l'autre. Ces deux fonctions sont échantillonnées et périodiques.

$$B_i(\nu) = T_S\left[I_i(t)\right] \qquad \text{d'après (67)}$$

La fonction réelle $B_i(\nu)$ est considérée comme la partie imaginaire de $B(\nu)$ dans l'expression (65) et $I_i(t)$ comme la partie impaire de $I(t)$ dans l'expression (64) en II.1.3. On peut donc écrire :

$$(106) \quad \begin{cases} B(\nu) = i\, B_i(\nu) \\ I(t) = I_i(t) \end{cases}$$

Dans la suite de cet exposé, nous ne parlerons plus que des deux fonctions $B(\nu)$ et $I(t)$ en sachant que $B(\nu)$ est impaire et imaginaire pure et que $I(t)$ est réelle et impaire.

avec : $B(\nu) = T^{+}\left[I(t)\right]$

Sur la figure 9.a, nous avons schématisé ces deux fonctions. $I(t)$ est périodique de période $2T$ et échantillonnée avec un pas $h = 1/2\Delta\nu'$. Le spectre $B(\nu)$ est périodique de période $2\Delta\nu'$ et échantillonné avec un pas $r = \frac{1}{2T}$. La période de $I(t)$ centrée sur l'origine $t=0$ représente l'interférogramme réel dont il faut calculer le spectre. Ce spectre se trouve dans une des demi-périodes (ou intervalle spectral libre) de la fonction $B(\nu)$.

Le pas d'échantillonnage r du spectre a été choisi de façon à ce que les conditions d'application du théorème d'échantillonnage (I.3) soient respectées. Il s'ensuit qu'il est possible d'isoler la période centrale de $I(t)$ représentant l'interférogramme en multipliant cette fonction par une "fonction rectangle" $D(t)$ définie par :

$$\begin{cases} D(t) = 1 \text{ si } -T \leq t \leq +T \\ \\ D(t) = 0 \text{ partout ailleurs} \end{cases} \qquad \text{voir figure 9.b} \tag{107}$$

On retrouve ainsi la fonction $I_4(t)$ étudiée en (I.4) qui est réelle limitée et échantillonnée, mais cette fois ci elle est de plus impaire.

$$I_4(t) = I(t).\ D(t) \qquad \text{voir figure 9.c} \tag{108}$$

Sa transformée $B_4(\nu)$ est une fonction continue périodique égale à $B(\nu)$ pour $\nu_k = k.r$

$$B_4(\nu) = T^+ \left[I_4(t) \right] = T^+ \left[I(t).D(t) \right]$$

Si $f(\nu)$ est la transformée de $D(t)$, nous aurons d'après (8) :

$$B_4(\nu) = B(\nu) \star f(\nu) \tag{109}$$

avec $$f(\nu) = \frac{\sin 2\pi\nu T}{2\pi\nu T} \qquad \text{voir figure 9.b}$$

L'expression (109) se développe sous la forme :

$$\boxed{B_4(\nu) = \sum_{k=-\infty}^{+\infty} B(\nu_k) \ \frac{\sin 2\pi T\ (\nu - \nu_k)}{2\pi T\ (\nu - \nu_k)}} \qquad \text{avec } \nu_k = k.r \tag{110}$$

Autrement dit, à partir des échantillons de $B(\nu)$, il est possible de définir par la somme infinie (110) n'importe quelle valeur du spectre continu $B_4(\nu)$ et ceci sans approximation.

Il est donc possible de définir par exemple un spectre secondaire interpolé $G(\nu)$ échantillonné avec un pas $r' = r/5$.

Nous aurons :

$$\begin{cases} G(kr) = B(kr) \\ \\ G(k'r') = B_4(k'r') \end{cases}$$

$G(\nu)$ peut donc être définie par :

(111) $G(\nu) = B_4(\nu).R_{r'}(\nu)$ figure 9.d

où $R_{r'}$ est une distribution de Dirac de période r'. Soit $I_5(t)$ la transformée de $G(\nu)$:

(112) $I_5(t) = T^{-}\left[G(\nu)\right] = I_4(t) \overset{*}{\underset{r'}{} } R_{\frac{1}{r'}}(t)$ figure 9.d

avec $1/r' = 10T$

La fonction $I_5(t)$ est donc périodique de période $10T$. De façon générale, on voit que le spectre secondaire $G(\nu)$ peut être défini avec un pas r' aussi petit qu'on veut. Si $r' \leqslant r$ la convolution représentée par (112) ne risque pas d'introduire les phénomènes de "recouvrement" d'une période sur l'autre qui risqueraient de distordre la partie de la fonction comprise entre $-T$ et $+T$.

III.1.2. - Principe de l'interpolation avec apodisation.

L'application de la formule d'interpolation (110) aux valeurs primaires $B(\nu_k)$ du spectre d'une fonction I(t) non pondérée définit les valeurs secondaires $G(\nu)$ du spectre non apodisé. Le calcul du spectre apodisé peut se faire de deux façons différentes α et β :

- α - Au lieu de partir de la fonction $I_4(t)$ telle qu'elle a été enregistrée expérimentalement, on commence par la multiplier par une fonction de pondération A(t). Les valeurs primaires $B'(\nu_k)$ du spectre apodisé $B'_4(\nu)$ qui en est la transformée sont distantes de r comme dans le cas précédent, mais sont différentes des valeurs $B(\nu_k)$. A partir des valeurs primaires $B'(\nu_k)$, on peut calculer n'importe quelle valeur du spectre continu apodisé $B'_4(\nu)$ en utilisant la convolution par la fonction $f(\nu)$ définie par l'expression (110).

- β - Repartons maintenant de la fonction I(t) non pondérée et de sa transformée de Fourier $B(\nu)$ représentées à nouveau sur la figure 10.a. Soit A(t) la

fonction de pondération de l'interférogramme dont la transformée est la fonction d'appareil (ou réponse percussionnelle) $f'(\nu)$ désirée. Si on convolue les valeurs primaires $B(\nu)$ du spectre non apodisé par la fonction $f'(\nu)$, on définit un nouveau spectre $B'_4(\nu)$ dont la transformée de Fourier est $I'_4(t)$ avec :

(113) $$B'_4(\nu) = B(\nu) \star f'(\nu)$$ fig. 10.b et c

(114) $$I'_4(t) = T^{-} \left[B'_4(\nu) \right] = I(t).A(t)$$

D'après (114), nous voyons que la convolution (113) définit une fonction $B'_4(\nu)$ continue et périodique qui est la transformée de la fonction $I(t)$ pondérée. Il s'agit donc bien du spectre apodisé. L'expression (113) peut s'écrire :

(115) $$\boxed{B'_4(\nu) = \sum_{k=-\infty}^{+\infty} B(\nu_k).f'(\nu - \nu_k)}$$ avec $\nu_k = k.r$

et $f'(\nu) = T_c \left[A(t) \right]$

Par cette expression on peut, à partir des valeurs de $B(\nu)$, calculer un échantillonnage secondaire $G'(\nu)$ cinq fois plus serré du spectre apodisé $B'_4(\nu)$. Fig. 10.d. Sa T.F. $I'_5(t)$ sera comme $I_5(t)$ une fonction périodique de période $10T$.

$$G'(\nu) = B'_4(\nu).R_{r'}(\nu) \quad \underline{\text{D'où}} : \quad I'_5(t) = I'_4(t) \star R_{\frac{1}{r'}}(t)$$

EN RESUME :

Pour obtenir des valeurs interpolées du spectre apodisé d'une fonction $I(t)$, on peut :

- soit calculer les valeurs primaires du spectre de la fonction $I(t)$ pondérée, puis interpoler ces valeurs par une fonction $f(\nu)$ de type $\sin x/x$.

- soit calculer les valeurs primaires du spectre de la fonction $I(t)$ non pondérée, puis interpoler ces valeurs par la fonction d'appareil $f'(\nu)$ désirée qui est en fait la T.F. de la fonction de pondération.

La seconde technique est préférable puisqu'à partir des valeurs primaires $B(\nu_k)$ non apodisées, il est possible de calculer un spectre interpolé pour lequel on pourra faire varier l'apodisation et ceci sans avoir à recalculer la DFT. C'est celle que nous avons adoptée dans nos programmes.

III.2. - Réalisation pratique de l'interpolation.

Les expressions (110) et (115) définissent une interpolation idéale mais irréalisable puisque les sommes portent sur une infinité de termes. En calcul numérique, nous devons donc les remplacer par une somme finie de ce type :

(116)
$$\begin{cases} B'_4(\nu) = \sum_{k=k_1-m}^{k_1+m} B(\nu_k).f(\nu - \nu_k) \\ \\ \text{avec } \nu_k = k.r \quad \text{et} \quad |k_1.r-\nu|<r \end{cases}$$

Dans cette expression, nous avons toujours :

$$|\nu - \nu_k| \leq m.r$$

Il revient donc au même de dire qu'on limite la somme comme en (116) à 2m+1 termes ou qu'on utilise la sommation infinie de l'expression (115) mais en limitant la fonction d'appareil $f(\nu)$ au domaine de fréquence $\left[-\nu_L , +\nu_L\right]$ avec :

(117) $$\nu_L = m.r$$

Pour diminuer la durée du calcul des valeurs interpolées on a intérêt à prendre la plus petite valeur possible pour m, mais la limitation correspondante du domaine de $f(\nu)$ introduit des erreurs sur les valeurs ainsi calculées. En effet, limiter le domaine de $f(\nu)$ revient à multiplier cette fonction par une "fonction rectangle" $D_f(\nu)$ de largeur 2m.r centrée sur f(0). Or si le filtre D(t) (ou A(t) en cas d'apodisation) correspondant à la fonction $f(\nu)$ non limitée permet d'isoler parfaitement la période de I(t) centrée sur l'origine, il n'en est pas de même du filtre D''(t) correspondant à la fonction $f(\nu)$ limitée.

(118) $f(\nu) \xrightarrow{T_c} D(t)$ filtre parfait

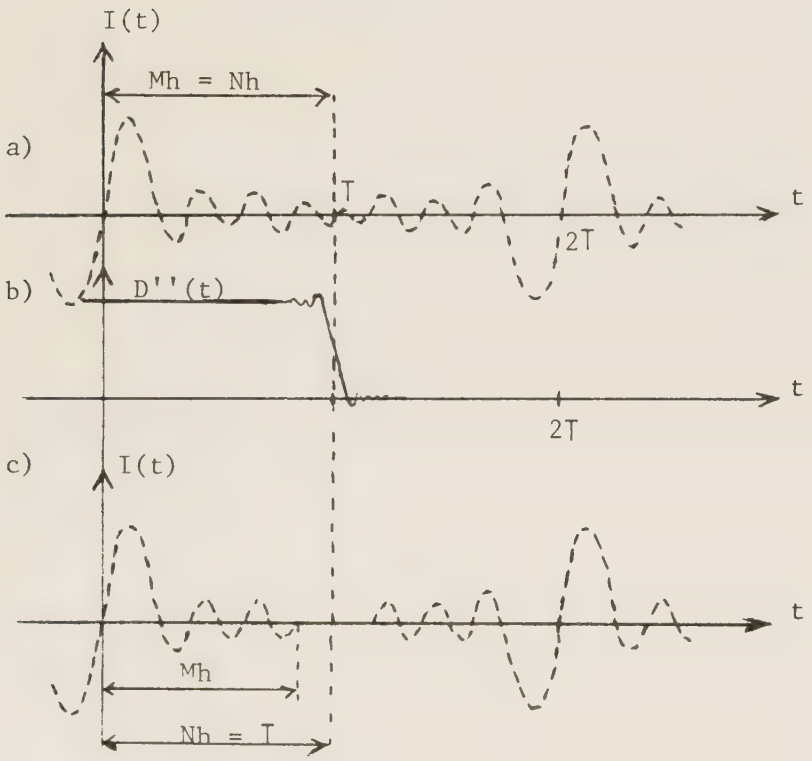
$f(\nu).D_f(\nu) \xrightarrow{T_c} D''(t)=D(t)*T_c\left[D_f(\nu)\right]$ filtre imparfait

Pour diminuer les erreurs d'interpolation liées à l'imperfection de ce filtre D''(t), il faut prendre deux précautions que nous allons étudier maintenant.

III.2.1. - Précaution concernant le calcul des valeurs primaires de B(ν).

Si on a enregistré M échantillons d'une fonction impaire $I_4(t)$ à partir de l'origine et si on prend $T=M.h$, le pas d'échantillonnage de son spectre tel qu'il est défini par le théorème d'échantillonnage sera : $r = \frac{1}{2Mh} = \frac{1}{2Nh} = \frac{1}{2T}$. La transformée de Fourier I(t) de ce spectre échantillonné est périodique de période 2T et elle est impaire par rapport au point d'abscisse T. Autrement dit les valeurs de I(t) au voisinage de l'abcisse T ne sont pas nulles. Or, c'est justement dans cette région que le filtre D''(t) est le plus déformé par rapport au filtre idéal D(t) dont les valeurs sont nulles pour t > T. Pour diminuer les effets de ce mauvais filtrage, on a alors intérêt à prendre dans le spectre un échantillonnage légèrement plus serré que ne le prévoit la théorie. Par exemple, au lieu de prendre :

$$r = \frac{1}{2Mh}$$



On prendra :

(119) $$r = \frac{1}{2Nh} \quad \text{avec} \quad M \simeq 0{,}98.N$$

Ceci revient à complèter les M valeurs de la fonction $I_4(t)$ à transformer par (N-M) zéros. Etant donné la symétrie de I(t), par rapport à l'abcisse T = N.h, il y aura (N-M) zéros de part de d'autre de cette abcisse (fig.c) et les imperfections du filtre dans cette région n'introduiront plus d'erreurs. Après filtrage par D''(t), on retrouvera ainsi les valeurs initiales de $I_4(t)$ avec une bien meilleure précision. Quand on utilise un programme "FFT" pour calculer les valeurs primaires du spectre, N est obligatoirement la puissance de 2 qui est immédiatement supérieure ou égale à M, ce qui fait qu'on est de toute façon amené à complèter la fonction avec des zéros. Dans tous les cas où M est voisin de N (voir (119)), ce qui pourrait passer pour un inconvénient de la méthode "FFT" peut donc, en fait, se révéler utile si on prévoit une interpolation ultérieure des valeurs primaires calculées.

III.2.2. - Pondération de la fonction d'interpolation f(ν).

Pour limiter la fonction f(ν) au domaine $[-\nu_L, +\nu_L]$ défini par (117), nous l'avions multipliée par une simple "fonction rectangle" $D_f(\nu)$. Les déformations correspondantes du filtre idéal étaient importantes puisque ce dernier était convolué par une fonction de type sin x/x dont les oscillations latérales ne s'amortissent pas rapidement. Il est donc préférable de remplacer $D_f(\nu)$ par une fonction de pondération $A_f(\nu)$. Le filtre idéal sera alors convolué par la transformée de $A_f(\nu)$. C'est-à-dire une fonction a(ν) dont les oscillations sont fortement atténuées. Une abondante littérature a été consacrée à l'amélioration du filtre par ce procédé.[4] [7] [16]

III.2.3. - L'algorithme du programme d'interpolation.

Reprenons l'expression (116) définissant le calcul numérique de l'interpolation :

$$B'_4(\nu) = \sum_{k=k_1 - m}^{k_1 + m} B(\nu_k) f(\nu - \nu_k)$$

avec $|k_1 r - \nu| < r$ et $\nu_k = kr$

La fonction d'appareil f(ν) qui joue le rôle de fonction d'interpolation dans cette formule est limitée au domaine $[-\nu_L, +\nu_L]$ avec :

$$\nu_L = mr = 10r$$

Nous verrons plus loin les détails concernant le calcul de la fonction $f(\nu)$, mais nous voyons que s'il s'agit d'une fonction de type sin x/x, nous n'en conserverons que 10 lobes latéraux de chaque coté du pic central. A partir des valeurs primaires $B(\nu_k)$, nous allons calculer un échantillonnage cinq fois plus serré. Soit $G(q.r')$ les valeurs de cet échantillonnage. Nous avons :

$$(120) \qquad G(qr') = B'_4(qr') = \sum_{k_1 - m}^{k_1 + m} B(\nu_k) f(qr' - \nu_k)$$

avec : $\nu_k = kr$; $m = 10$; $r' = \frac{r}{5}$

Si on remarque que qr peut s'exprimer sous la forme :

$$(121) \qquad qr = k_1 r + k_2 r' \qquad \text{avec} \quad 0 \leq k_2 < 5$$

L'expression (120) devient :

$$(122) \qquad G(k_1 r + k_2 r') = \sum_{k=k_1 - 10}^{k_1 + 10} B(kr) f(k_1 r + k_2 r' - kr)$$

Si on pose : $k' = k_1 - 11$, on obtient :

$$(124) \qquad G\{(k' + 11) r + k_2 r'\} = \sum_{k=k' + 1}^{k' + 21} B(kr) f\{(k' + 11) r + k_2 r' - kr\}$$

et en posant : $k = k' + n$:

(125)

$$G\{(k' + 11)r + k_2 r'\} = \sum_{n=1}^{21} B\{(k'+ n)r\} f\{k' + 11)r + k_2 r' - (k' + n)r\}$$

En tenant compte du fait que $f(\nu)$ est une fonction paire :

$$(126) \qquad \boxed{G\{(k' + 11)r + k_2 r'\} = \sum_{n=1}^{21} B\{(k' + n)r\} f\{(5n - k_2 - 55)r'\}}$$

Soit k'_0 la première valeur possible de k' et supposons que k_2 soit nul. Nous aurons alors :

$$G\{(k'_0 + 11)r\} = \sum_{n=1}^{21} B\{(k'_0 + n)r\} f\{(5n - 55)r'\} \tag{127}$$

La première valeur de $B(\nu)$ qui sera utilisée dans ce calcul sera : $B(\nu_0) = B\{k'_0 + 1)\ r\}$. Les valeurs primaires de $B(\nu)$ que nous traitons dans nos programmes sont stockées sur bande magnétique par blocs de 1024 valeurs. Ces blocs seront traités les uns après les autres après avoir été lus sur la bande et stockés en mémoire centrale dans un tableau B de 1044 mots. Nous justifierons plus loin le choix de cette valeur supérieure à 1024. Soit $\{B_i\}$ l'ensemble des 1044 valeurs primaires stockées avec :

$$\begin{cases} B_1 = B(\nu_0) \\ B_2 = B(\nu_0 + r) \\ \vdots \\ B_i = B\{\nu_0 + (i-1)\,r\} \end{cases} \tag{128}$$

La première valeur calculée de $G(\nu)$ sera $G(\nu_0 + 10r)$ et l'ensemble $\{G_k\}$ des 5120 valeurs interpolées sera stocké dans un tableau G de 5120 mots avec :

$$\begin{cases} G_1 = G(\nu_0 + 10r) = G\ (\nu'_0) \\ G_2 = G(\nu'_0 + r') \\ \vdots \\ G_k = G\ \{\nu'_0 + (k-1)\,r'\} \end{cases} \tag{129}$$

Nous pouvons remarquer, que dans le spectre, la valeur interpolée G_1 remplacera la valeur primaire B_{11}.

La fonction $f(\nu)$ doit être échantillonnée avec un pas r'. D'après (126) et (121), nous pouvons voir entre quelles limites varie l'abcisse de f soit :

$$(5n - k_2 - 55)\ r' = \begin{cases} -\ 54\ r' & \text{si } n = 1 \text{ et } k_2 = 4 \\ 0 & \text{si } n = 11 \text{ et } k_2 = 0 \\ +\ 50\ r' & \text{si } n = 21 \text{ et } k_2 = 0 \end{cases}$$

Nous avons donc stocké les valeurs de $f(\nu)$ dans un tableau A de 109 mots en mémoire centrale. Soit $\{A_j\}$ l'ensemble de ces 109 valeurs avec :

$$(130)\quad \begin{cases} A_{55} = f(0) \\ A_{56} = A_{54} = f(r') \\ \vdots \quad \vdots \quad \vdots \\ A_j = A_{110-j} = f(j-55)r' \\ \vdots \quad \vdots \quad \vdots \\ A_{109} = A_1 = f(54r') \end{cases}$$

En fonction de ces nouvelles notations, l'expression (126) peut s'écrire :

$$(131)\quad G_k = \sum_{n=1}^{21} B_i\, A_j$$

Il nous reste alors à définir les relations qui existent entre les indices k, i et j. D'après (129), nous avons :

$$G_k = G\,\{\nu_0 + 10r + (k-1)\,r'\}$$

Si $E\left[x\right]$ est la partie entière de la valeur x, nous pouvons développer l'argument de G_k sous la forme :

$$G_k = G\,\{\nu_0 + 10r + E\left[\frac{k-1}{5}\right]r + k_2\,r'\} = G\,(\nu_G)$$

Dans l'expression (126), nous voyons que l'abcisse ν_B des valeurs $B(\nu)$ à utiliser dans la somme, se déduit de l'abcisse ν_G de la valeur G_k par :

$$\nu_B = \nu_G - 11r - k_2 r' + nr$$

En fonction de n, l'expression de B_i sera donc :

$$B_i = B\,\{\nu_0 + (E\left[\frac{k-1}{5}\right] + n-1)\,r\}$$

Si on la compare à l'expression générale de B_i définie en (128), on obtient la relation :

$$(132)\quad i = E\left[\frac{k-1}{5}\right] + n$$

Voyons maintenant comment s'exprime l'abcisse ν_f de la fonction $f(\nu)$ dans l'expression (126) :

$$\nu_f = (5n - k_2 - 55)r'$$

En comparant à l'abcisse générale de f_j définie en (130), on voit que :

$$(133) \qquad j = 5n - k_2 = 5n - \left(k-1 - 5E\left[\frac{k-1}{5}\right]\right)$$

Les expressions (131), (132), (133) reliées aux ensembles de valeurs $\{G_k\}$, $\{B_i\}$, $\{A_j\}$ stockés dans les tableaux G, B et A définissent l'algorithme permettant le calcul des valeurs interpolées du spectre :

$$(134) \qquad \boxed{\begin{array}{l} G_k = \sum_{n=1}^{21} B_i\, A_j \qquad \text{avec } k=1,2,\ldots,5120 \\ \text{pour } k \text{ donné, on a : } \begin{cases} i = k_1 + n \\ j = 5n - k_2 \end{cases} \text{ avec : } \begin{cases} k_1 = E\left[\frac{k-1}{5}\right] \\ k_2 = k-1 - 5k_1 \end{cases} \end{array}}$$

III.2.3. α Choix de la dimension du tableau B des valeurs primaires.

Nous pouvons justifier maintenant le choix de la valeur 1044 pour la dimension du tableau B. Le but de l'interpolation dans le cas particulier de nos spectres, est de faire correspondre à chaque bloc de 1024 valeurs primaires lu sur bande magnétique, cinq blocs de valeurs interpolées, c'est-à-dire 5120 valeurs de $G(\nu)$. Pour que le calcul par (134) d'une valeur $G(\nu_1)$ soit possible, il faut pouvoir disposer d'au moins dix valeurs primaires de chaque coté de l'abcisse ν_1. C'est dans ce but que nous avons dit que G_1, la première valeur calculée, correspondait à l'abcisse de B_{11}. Il en sera de même pour la dernière valeur calculée soit G_{5120}. En effet pour k=5120, la fonction d'appareil $f(\nu)$ "s'appuie" sur 21 valeurs primaires B_i et la plus grande valeur de l'indice i peut être calculée d'après (134) en prenant n=21, ce qui donne :

$$i = E\left[\frac{5119}{5}\right] + 21 = 1044$$

Il est facile de vérifier d'autre part que :

$$\text{Abcisse de } G_{5120} = \text{Abcisse de } B_{1034} - r'$$

Il faudrait donc en principe ajouter 10 valeurs primaires à chaque extrémité du bloc de 1024 valeurs lu sur bande. Pour simplifier, on complète en fait ce bloc à 1044 valeurs en lui ajoutant les 20 valeurs qui le précèdent. Lorsqu'il s'agit d'un bloc intermédiaire dans le spectre, on ajoute les 20 valeurs venant de la fin du bloc précédent. Lorsqu'il s'agit du premier bloc, on peut ajouter 20 valeurs en tenant compte du fait que la fonction $B(\nu)$ est impaire par rapport à l'abcisse du premier échantillon qui se trouve au début de l'intervalle spectral libre. En conclusion, si ν_{1B} est l'abcisse de la première valeur du bloc lu, l'abcisse ν_{1G} de G_1 sera :

(135) $$\nu_{1G} = \nu_{1B} - 20r + 10r = \nu_{1B} - 10r$$

III.2.3. β Calcul de la fonction d'interpolation $f(\nu) \rightarrow \{A_j\}$

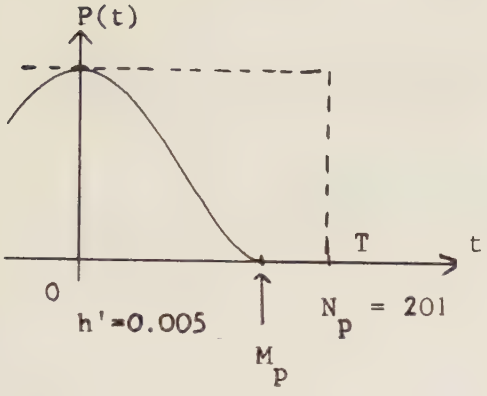
Comme nous l'avons vu en III.1.2, il est préférable de ne pas pondérer directement les valeurs de $I_4(t)$. Nous apodisons donc toujours le spectre au moment de l'interpolation. Si on ne désire pas apodiser, la fonction d'interpolation $f(\nu)$ sera une fonction $\frac{\sin x}{x}$ qui est la transformée de Fourier du filtre idéal rectangulaire isolant la période centrale de I(t) (figure 9 a et b) Si on veut apodiser le spectre, la fonction d'interpolation sera calculée comme transformée du filtre pondéré A(t) (figure 10 a et b). Nous avons vu en III.2.1 que le nombre M de valeurs de $I_4(t)$ était en général inférieur à N la puissance de 2 utilisée pour le calcul de la FFT. Dans ce cas, la fonction de pondération doit être définie sur ces M valeurs. Il faut donc calculer 50 valeurs de la transformée $f(\nu_k)$ de ces fonctions filtre pour les fréquences ν_k avec :

$$\nu_k = kr' \qquad \text{avec} \quad r' = \frac{1}{10Nh}$$

$$0 \leqslant k < 50$$

Ce calcul doit être fait par une DFT classique et pour les grandes valeurs de M, il serait trop long de calculer chaque valeur $f(\nu_k)$ à partir des M (ou N) valeurs du filtre. On définit donc un filtre rectangulaire de longueur $2T = 2$ et on l'échantillonne avec 201 valeurs prises à partir de l'origine t=0. Le pas h' d'échantillonnage est donc égal à $T/200=0.005$.

S'il faut apodiser le spectre, on pondère ce rectangle par une fonction P(t) en réalisant une similitude entre la pondération des N valeurs réelles de $I_4(t)$ et les 201 valeurs de ce rectangle. Quelle que soit la fonction P(t) choisie, ses valeurs P_k seront telles que :



$$P_k = P\left[(k-1)h'\right] \quad \text{avec } k=1,2,..,N_p=201$$

et on aura toujours : $P_1 = P(0) = 1$

et $P_k = 0$ pour $k \geq M_p$, M_p étant défini par : $\frac{M_p}{N_p} = \frac{M}{N}$

On obtient ainsi un ensemble $\{P_k\}$ de M_p valeurs dont on calcule la DFT. $f(\nu_j)$ pour des fréquences ν_j définies par :

$$\nu_j = \frac{j-1}{5} \times \frac{1}{2T} = (j-1) \times 0,1 \qquad \text{pour } j = 1,2,...,50$$

Les valeurs $f(\nu_j)$ ainsi calculées constituent les valeurs A_{j+54} du tableau A.

Ces 50 valeurs sont alors pondérées (voir III.2.2) par une fonction du type : $\left(1 - \frac{\nu_j^2}{\nu_{50}^2}\right)^2$

REMARQUES

- *Cas particulier de l'interpolation sans apodisation.*

Nous avons vu que si on ne profite pas de l'interpolation pour apodiser le spectre, la fonction d'interpolation $f(\nu)$ est une simple fonction sin x/x calculée pour des valeurs ν_k avec :

$$f(\nu) = \frac{\sin \pi\nu 2T}{\pi\nu 2T} = T_c\left[D_{2T}(t)\right]$$

et $\nu_k = kr'$ pour $k = 0,1,2,...,50$

Comme : $r' = \frac{r}{5} = \frac{1}{10\bar{T}}$, $f(\nu_k)$ peut s'écrire :

$$f(\nu_k) = \frac{\sin 2\pi \frac{k}{10}}{2\pi \frac{k}{10}}$$

Ces valeurs $f(\nu_k)$ sont indépendantes de la largeur réelle $2\bar{T}$ du filtre rectangulaire $D_{2\bar{T}}(t)$. Si elles ont été calculées une fois pour toutes, puis stockées dans un tableau du programme, elles peuvent être utilisées pour interpoler n'importe quel spectre lorsque aucune apodisation ne doit être introduite à ce niveau. On peut ainsi diminuer la durée du calcul et améliorer la précision sur les valeurs $f(\nu_k)$. A ce propos, on peut remarquer que les valeurs $G(\nu)$ interpolées qui correspondent à l'abcisse ν d'une valeur primaire $B(\nu)$ seront exactement égales à ces dernières. En effet, reprenons l'expression (127) qui définissant justement le calcul d'une de ces valeurs :

$$G\{(k'_0 + 11)r\} = \sum_{n=1}^{21} B\{(k'_0 + n)r\} f\{(5n - 55)r'\}$$

$$\begin{cases} \text{Pour } n = 11 & f \text{ est égale à } 1 \\ \text{Pour } n \neq 11 & f \text{ est nulle} \end{cases}$$

D'où : $G\{(k'_0 + 11)r\} = B\{(k'_0 + 11)r\}$

Justification du choix de la valeur $N_p = 201$

Cette valeur représente un compromis entre la précision du calcul des valeurs $f(\nu_j)$ et la durée du calcul qui est proportionnel à $50\ N_p$. L'intervalle spectral libre lié au pas h' est : $\Delta\nu = 1/2h' = 100$. La largeur des lobes de la fonction $\frac{\sin x}{x}$ calculée étant de $\frac{1}{2\bar{T}} = 0.5$, il y aura 200 lobes dans l'intervalle spectral libre ce qui permet à la fonction $\frac{\sin x}{x}$ de s'atténuer et de devenir presque négligeable au voisinage du point de recouvrement $\Delta\nu'$.

Cette dernière précaution peut devenir insuffisante si $M \ll N$ car dans ce cas, la largeur des lobes de la fonction d'appareil apodisée s'accroît. Ce facteur joue d'une part sur le problème de recouvrement et donc sur la précision des valeurs $f(\nu_j)$ calculées, mais aussi sur la précision des valeurs

interpolées car les 50 valeurs calculées correspondent à beaucoup moins de 10 lobes de f(ν). Le cas M << N peut se présenter dans le cas où, au moment de l'interpolation, on veut diminuer la résolution dans le spectre.

Différents types d'apodisation

Suivant la valeur du paramètre "IAP" dans les programmes, il est possible de choisir parmi cinq types de fonctions d'appareil que nous allons définir par la forme de la fonction de pondération P(t) qui leur correspond.

Soit P_k la valeur de P(t) appliquée à la $k^{ième}$ valeur du rectangle de N_p valeurs pour le pondérer jusqu'à la $M_p^{ième}$ valeur.

Posons :

$$\begin{cases} F = 1 - \dfrac{(k-1)^2}{(M_p-1)^2} & \text{pour } 1 \leq k \leq M_p \\ F = 0 & \text{pour } M_p < k \leq N_p \end{cases}$$

- Si "IAP" = 0 → Pas d'apodisation (SA) $P_k = 1 \ \forall_k$.

- Si "IAP" = 1 → Apodisation de type "APO1" $P_k = F^2$

- Si "IAP" = 2 → Apodisation de type "APO2"
  $P_k = 0{,}0627+0{,}3221F+0{,}2184F^2+0{,}3868F^3$

- Si "IAP" = 3 → Apodisation de type "APO3"
  $P_k = 0{,}2398+0{,}4581F+0{,}2250F^2+0{,}772F^3$

- Si "IAP" = 4 → Apodisation de type "APO4" $P_k = F^{5/2}$

La forme de la fonction d'appareil pour chacune de ses apodisations est représentée sur la figure 11.

Programme d'interpolation et de calcul de f(ν).

L'expression (134) peut être programmée à l'aide du groupe n°1 d'instructions, ci-dessous. Pour gagner du temps, on peut développer la boucle principale comme dans le groupe n°2 et gagner ainsi un facteur qui est compris entre 1,6 et 4,6 suivant le compilateur utilisé.

Le calcul de la fonction d'interpolation se trouve dans le groupe 3 d'instructions.

Correspondances des notations (celles du programme sont à droite)

(134)

k → K	f ou A → AA	ν_{1B} → SIG1B
i → NZ ou I	F → F	r → DELTAB
j → KZ ou J	h → DELTAD	r' → DELTAG
G → G	M → M	ν_{1G} → SIG1G
B → B	M_P → MP	n → N
		P_k → PK
		N → NPRIM

GROUPE 1

```
      DO 193 KB=1,NBLOC
      READ(16)(B(J),J=21,1044)
C        INTERPOLATION.
C        ----------------
      DO 194 K=1,5120
      G(K)=0.
      I=(K-1)/5
      J=I*5-K+1
      DO 194 N=1,21
      I=I+1
      J=J+5
  194 G(K)=G(K)+B(I)*AA(J)
C
      DO 196 I=1,20
  196 B(I)=B(I+1024)
  193 CONTINUE
```

GROUPE 2

```
C
C     CONVOLUTION ET INTERPOLATION.
      DO 194 K=1,5120
      I=(K-1)/5+1
      J=I*5-K+1
  194 G(K)=B(I)*AA(J)+B(I+20)*AA(J+100)+B(I+1)*AA(J+5)+B(I+19)*AA(J+95)+
     1B(I+2)*AA(J+10)+B(I+18)*AA(J+90)+B(I+3)*AA(J+15)+B(I+17)*AA(J+85)+
     2B(I+4)*AA(J+20)+B(I+16)*AA(J+80)+B(I+5)*AA(J+25)+B(I+15)*AA(J+75)+
     3B(I+6)*AA(J+30)+B(I+14)*AA(J+70)+B(I+7)*AA(J+35)+B(I+13)*AA(J+65)+
     4B(I+8)*AA(J+40)+B(I+12)*AA(J+60)+B(I+9)*AA(J+45)+B(I+11)*AA(J+55)+
     5B(I+10)*AA(J+50)
      DO 196 I=1,20
  196 B(I)=B(I+1024)
C
```

GROUPE 3

```
C
C *** CALCUL DE LA FONCTION D'INTERPOLATION AA.
C-------------------------------------------------
C *** AA=DFT. D'UN RECTANGLE DE  LARGEUR 2T=2 ECHANTILLONNE AVEC 201 VALEURS SUR
C *** (0,T). (DELTAD=0.005). L'ARGUMENT DU COSINUS DANS LA DFT. EST EGAL A
C *** 2*PI*DELTA*SIG AVEC DELTA=FIK*0.005 ET SIG=FIJ/(2*T*5)=0.1*FIJ.
C *** SI IAP N'EST PAS NUL,CE RECTANGLE PEUT ETRE PONDERE SUR MP VALEURS.
C
      DO 1 I=2,200
    1 B(I)=1.0
      B(201)=0.5
C *** PONDERATION DE L'INTERFEROGRAMME.
      IF(IAP.EQ.0)GO TO 12
      MP=(201*M)/NPRIM
      Q=(MP-1)*(MP-1)
      IF(IAP-2)1935,1936,1937
 1935 R=0.                                                          AP01
      S=0.
      T=1.
      W=0.
      V=2.
      GO TO 5000
 1936 R=0.06267                                                     AP02
      S=0.33213
      T=0.21839
      W=0.38680
      V=2.
      GO TO 5000
 1937 IF(IAP-3)1939,1939,1940
 1939 R=0.23978                                                     AP03
      S=0.45808
      T=0.22499
      W=0.077208
      V=2.
      GO TO 5000
 1940 R=0.                                                          AP04
      S=0.
      T=1.
      W=0.
      V=3./2.
 5000 DO 9 I=2,MP
      FI=I-1
      F=1.-(FI*FI)/Q
      PK=R+S*F+T*F**V+W*F**3
    9 B(I)=B(I)*PK
C *** CALCUL TF CLASSIQUE  AA=TF.(B)
   12 AMAX=0.5
      DO 4 IK=2,MP
    4 AMAX=AMAX+B(IK)
      SIA=+0.3141592650-02
      DO 7 IJ=56,105
      FIJ=IJ-55
      SIG=SIA*FIJ
      AA(IJ)=0.5
      DO 3 IK=2,MP
      FIK=IK-1
    3 AA(IJ)=AA(IJ)+B(IK)*COS(SIG*FIK)
C *** NORMALISATION ET PONDERATION DE LA FONCTION AA.
    7 AA(IJ)=(AA(IJ)/AMAX)*(1.-(FIJ*FIJ)/2500.)**2
      AA(55)=1.
      DO 8 I=106,109
    8 AA(I)=0.
C *** DEFINITION PAR SYMETRIE DE LA FCT. D'INTERP. AA COMPLETE
      DO 181 I=1,54
  181 AA(I)=AA(110-I)
```

III.3. - RESULTATS NUMERIQUES

Pour évaluer l'incertitude sur le calcul des valeurs interpolées, nous avons fait une série de tests (voir tableaux 1, 2 et 3 page suivante), concernant l'erreur maximum relevée lors de l'interpolation d'une fonction constante normalisée à 1.

Cette restitution a été faite avec et sans apodisation. Dans le deuxième cas (IAP=1, voir III.2.3.β p.75), les valeurs interpolées ont été renormalisées à 1 avant de relever l'erreur maximum. Nous avons étudié l'influence de trois facteurs :

m : Chaque point interpolé est calculé à partir de 2 m valeurs primaires (voir (120) p. 68).

N_p : La fonction d'interpolation $f(\nu)$ est la T.F. d'un filtre P(t) échantillonné avec 2 N_p valeurs (voir III.2.3.β p.73).

F_p : La fonction $f(\nu)$ est pondérée (III.2.2 p.67) par une fonction du type $(1. - \frac{\nu^2}{\nu^2_{Max}})^{F_p}$

Pour l'exploitation courante en spectroscopie de Fourier, nous avons choisi comme valeurs :

$m = 10$, $N_p = 201$, $F_p = 2$.

TABLEAU 1 - INFLUENCE DE N_p

Deux raisons nous amènent à prendre une valeur aussi faible que possible pour N_p. La première est que la fonction d'interpolation étant calculée par une D.F.T. classique à partir des N_p valeurs de P(t), la durée de ce calcul sera proportionnelle à N_p. La deuxième est liée au fait qu'un calcul de D.F.T. classique à partir d'un grand nombre de valeurs nécessiterait l'utilisation de la double précision pour être valable (voir $N_p = 1001$ et IAP = 1).

Nous avons vu en III.2.3.β p.74 les inconvénients qu'il y aurait à prendre une trop faible valeur de N_p surtout dans le cas où il y a apodisation.

REMARQUE :

Pour $N_p = 201$ et $m = 10$, la durée du calcul de $f(\nu)$ est de 20 $1/100^{ème}$ de seconde sur un 370/165 IBM.

INFLUENCE DU PARAMETRE "N_p" ($m = 10$, $F_p = 2$)

N_p	IAP=0	IAP=1	IAP=1 double préc.
1001	$2.0\ 10^{-4}$	$7.0\ 10^{-5}$	$1.5\ 10^{-6}$
501	$2.0\ 10^{-4}$	$1.5\ 10^{-6}$	$1.5\ 10^{-6}$
201	$2.8\ 10^{-4}$	$0.5\ 10^{-6}$	$1.5\ 10^{-6}$
101	$2.8\ 10^{-4}$	$2.5\ 10^{-6}$	$2.0\ 10^{-6}$

TABLEAU 1

INFLUENCE DU PARAMETRE "m" ($N_p = 201$, $F_p = 2$)

m	IAP=0	IAP=1
30	$5.0\ 10^{-6}$	$2.5\ 10^{-6}$
20	$3.0\ 10^{-5}$	$2.5\ 10^{-6}$
10	$2.8\ 10^{-4}$	$5.0\ 10^{-7}$
5	$2.3\ 10^{-3}$	$4.5\ 10^{-5}$

TABLEAU 2

INFLUENCE DU PARAMETRE "F_p" ($N_p = 201$, $m = 10$)

F_p	IAP=0	IAP=1
3.	$4.0\ 10^{-6}$	$2.0\ 10^{-6}$
2.5	$8.4\ 10^{-5}$	$1.5\ 10^{-6}$
2.0	$2.8\ 10^{-4}$	$0.5\ 10^{-6}$
1.5	$5.3\ 10^{-4}$	$0.5\ 10^{-6}$
0	$3.0\ 10^{-2}$	$0.4\ 10^{-4}$

TABLEAU 3

★ Pour ces trois tableaux, les valeurs indiquées correspondent à l'erreur maximum relevée après interpolation d'une fonction constante normalisée à 1.

VALEURS CALCULEES DE LA FONCTION D'INTERPOLATION f(ν). $[A_{55} \rightarrow A_{105}]$ (voir (129))

IAP = 0

```
 1.000000  0.934740  0.754404  0.500921  0.230885
 0.000000 -0.151452 -0.207834 -0.179634 -0.097310
-0.000000  0.077004  0.112012  0.101211  0.056745
 0.000000 -0.047097 -0.069626 -0.063691 -0.036033
-0.000000  0.030206  0.044723  0.040889  0.023076
 0.000000 -0.019142 -0.028116 -0.025453 -0.014196
-0.000000  0.011427  0.016474  0.014598  0.007944
 0.000000 -0.006021 -0.008363 -0.007099 -0.003674
-0.000000  0.002445  0.003119  0.002383  0.001080
 0.000000 -0.000479 -0.000436 -0.000193 -0.000030
 0.0
```

IAP = 1 (M_p = 184)

```
 1.000000  0.975826  0.906058  0.798528  0.664928
 0.519030  0.374687  0.243944  0.135597  0.054399
 0.000996 -0.027484 -0.036358 -0.032101 -0.021099
-0.008659  0.001576  0.007785  0.009744  0.008359
 0.005088  0.001416 -0.001526 -0.003144 -0.003384
-0.002585 -0.001278  0.000016  0.000928  0.001305
 0.001191  0.000764  0.000241 -0.000195 -0.000438
-0.000475 -0.000359 -0.000175 -0.000002  0.000109
 0.000145  0.000122  0.000070  0.000019 -0.000013
-0.000024 -0.000019 -0.000009 -0.000002  0.000000
 0.0
```

(N_p = 201, m = 10, F_p = 2)

TABLEAU 2 - INFLUENCE DE m

Les grandes valeurs de m sont à écarter puisque la durée du calcul d'interpolation est proportionnelle à la valeur de ce paramètre. L'influence de ce paramètre est plus grande dans le cas où il n'y a pas d'apodisation puisqu'on est alors amené à tronquer une fonction sin x/x dont les pieds sont plus importants. De façon générale, nous avons vu en III.2.(118) que les valeurs trop faibles de m entraînent une distorsion de la forme du filtre P(t).

TABLEAU 3 - INFLUENCE DE F_p

Dans nos tests, la fonction d'interpolation f(ν), dont le domaine est théoriquement infini, est pondérée par :

$$A_f(\nu) = \left(1 - \frac{\nu^2}{\nu_L^2}\right)^{F_p} \quad \text{avec} \quad \nu_L = mr \quad \text{voir (119) et (120).}$$

Nous avons vu en III.2.2 que le filtre idéal D(t) (ou A(t) en cas d'apodisation) dont f(ν) est la T.F. se trouvait alors convolué par a(ν) la T.F. de $A_f(\nu)$. Si $F_p = 0$, $A_f(\nu) = 1$ et le filtre idéal est convolué par une fonction a(ν) de type sin x/x. Le filtre sera alors d'autant plus déformé par les oscillations de a(ν) que ν_L sera petit. Ces oscillations sont atténuées en amplitude quand F_p augmente, et on constate bien sur nos tests que la précision de l'interpolation augmente alors qu'elle est désastreuse pour $F_p = 0$.

Lorsque F_p augmente trop, les oscillations de a(ν) sont bien atténuées, mais la largeur de son pic central et de ses lobes latéraux s'accroit ce qui introduit une nouvelle déformation du filtre idéal. Les effets de ce mauvais filtrage ne peuvent être mis en évidence par l'interpolation d'une fonction constante et n'apparaissent donc pas dans le tableau 3, mais ils sont apparus clairement dans d'autres tests portant sur l'interpolation de spectres de raies. Nous avons donc choisi la valeur souvent utilisée $F_p = 2$ qui correspond à un compromis optimum entre les valeurs trop faibles ou trop forte de ce facteur.

CONCLUSION

La bonne restitution d'un niveau constant constitue un test de précision nécessaire, mais non suffisant. Il permet néanmoins de choisir des valeurs cohérentes pour ces trois paramètres. Avec les valeurs choisies, l'erreur relative sur la restitution d'un niveau constant est toujours inférieure à 3.10^{-4} sans apodisation et inférieure à 10^{-5} avec apodisation. Des mesures de précision faites sur des flancs de raies en émission et en absorption montrent que l'erreur reste inférieure à 3.10^{-3} pour des raies normalisées à 1. Dans un essai particulier d'interpolation (sans apodisation) d'une raie d'émission d'intensité 1, l'erreur est restée inférieure à 10^{-4}. Ces dernières mesures ont été faites en comparant les valeurs interpolées aux valeurs calculées directement en double précision par une D.F.T. classique. L'interpolation (5 points secondaires par point primaire) d'un bloc de 1024 valeurs primaires prend 14 1/100ème de seconde sur un 370/165.

IV. - PROBLEMES SPECIFIQUES DE LA SPECTROSCOPIE PAR TRANSFORMATION DE FOURIER

IV.1. - Acquisition des données - Enregistrement et lecture.

IV.1.1. Généralités

Avec la première génération (1964-1966) d'interféromètres, il fallait enregistrer environ 60 000 points à une cadence d'échantillonnage maximum de 5 mesures par seconde, ce qui permettait l'utilisation du ruban de papier perforé. C'est sur ce support qu'ont été enregistrés les interférogrammes correspondant aux spectres planétaires rassemblés dans l'atlas de P. CONNES [37]. L'enregistrement de 64 000 valeurs, se faisait en 105 minutes environ. Le ruban perforé fut aussi utilisé pour l'interféromètre de J. PINARD [9] (1965-1968) qui était conçu pour l'étude spectrale à haute résolution mais sur un domaine de fréquences relativement étroit ($\simeq 400\ cm^{-1}$). Cette dernière caractéristique permettait de limiter le nombre de points à 64 000 en augmentant le pas d'échantillonnage.

Avec les derniers interféromètres conçus pour l'étude à haute résolution de larges domaines spectraux, le nombre de points à enregistrer se situe entre 10^6 et $2\ 10^6$ et la cadence d'échantillonnage aux environs de 100 points par seconde. L'enregistrement d'un tel volume de données à une telle cadence, ne peut se faire de façon pratique qu'en utilisant un enregistreur incrémental à bande magnétique. Comme nous allons le voir, deux types d'enregistreurs sont utilisés. Les uns, plus anciens, utilisent des bandes "7 pistes" (densité 556 bpi) les autres, et c'est la solution d'avenir, des bandes "9 pistes" (densité 800 bpi).

Le codage

Les interférogrammes dont nous enregistrons les échantillons sur bande magnétique présentent un certain nombre de caractéristiques physiques dont nous allons voir l'influence sur le type de codage adopté pour stocker l'information.

Une première caractéristique, liée à la largeur du domaine spectral et à la haute résolution obtenue est que l'ordre de grandeur des fluctuations

d'intensité de l'interférogramme autour de sa valeur moyenne varie énormément en fonction de la différence de marche. Pour augmenter la dynamique du système enregistreur, tout en n'utilisant que le minimum de chiffres significatifs par valeur, on est donc amené à utiliser un atténuateur à 4 ou 6 positions dont le gain pourra prendre par exemple les valeurs 1, 10, 100 et 1000. A chacune de ces positions, doit alors correspondre une indication de gamme dans le codage de la valeur.

Une deuxième caractéristique est que la valeur moyenne de l'interférogramme étant nulle, les points enregistrés ont des valeurs positives nulles ou négatives. Il faut donc prévoir le codage du signe de chaque point de l'interférogramme.

Compte tenu du fait qu'un signe et une gamme sont associés à chaque point, cinq chiffres significatifs (décimaux) sont suffisants pour obtenir une précision satisfaisante dans le spectre calculé.

Enfin une dernière caractéristique propre à la spectroscopie par T.F. est que toute erreur concernant une des valeurs de l'interférogramme a des répercussions sur l'ensemble des valeurs du spectre. S'il s'agit d'une valeur erronée, une sinusoïde d'autant plus importante que l'erreur est grande se superpose au spectre. En adoptant les conventions de la DFTN vues en I.5.2, toute erreur d'intensité sur le $k^{ième}$ échantillon de l'interférogramme se traduit par l'apparition d'une sinusoïde de période $T = N'/(k-1)$ dans le spectre où le pas d'échantillonnage est égal à 1.

S'il s'agit d'un point sauté, les conséquences sont plus catastrophiques car, lorsqu'on ne connait pas l'emplacement du point sauté, l'interférogramme est pratiquement inutilisable. Si au contraire on connait son emplacement, on peut le remplacer par une valeur nulle (valeur moyenne) et on est ainsi ramené au cas de la valeur erronée. Nous voyons donc qu'il est très important d'avoir un système enregistreur très fiable. Dans ce but, plusieurs améliorations ont été prévues. La première concerne le remplacement des enregistreurs sur 7 pistes déjà anciens par de nouveaux sur 9 pistes plus fiables techniquement. La deuxième est de profiter de ce changement de technique pour associer au codage de chaque valeur son numéro ou sa position dans l'interférogramme ce qui peut permettre de repérer un point sauté. Lorsqu'on lit sur la bande un bloc de valeurs par l'intermédiaire d'une instruction "READ" en FORTRAN, le compilateur I.B.M. génère en "ASSEMBLEUR" une série d'ins-

tructions qui se révèle tout à fait insuffisante dans le cas où, au cours de la lecture du bloc, une erreur "hardware" de type quelconque est détectée. En effet, dans ce cas, le bloc incriminé n'est en général, pas transmis et le message d'erreur édité ne permet pas un diagnostic précis. Le fait que le bloc ne soit pas transmis, risque d'entrainer un décalage dans la numérotation des blocs suivants ce qui, comme nous l'avons vu, est catastrophique pour un interférogramme. De plus, il est inadmissible d'ignorer toutes les valeurs d'un bloc dans le cas où, en fait, seules quelques valeurs sont "illisibles" du fait de l'erreur "hardware". Une troisième amélioration a donc consisté à remplacer l'instruction "READ" par l'appel à un sous-programme écrit en "assembleur" qui permet justement de récupérer ce bloc, et d'en donner son image en mémoire de façon à améliorer le diagnostic d'erreur et le décodage éventuel des valeurs correctes.

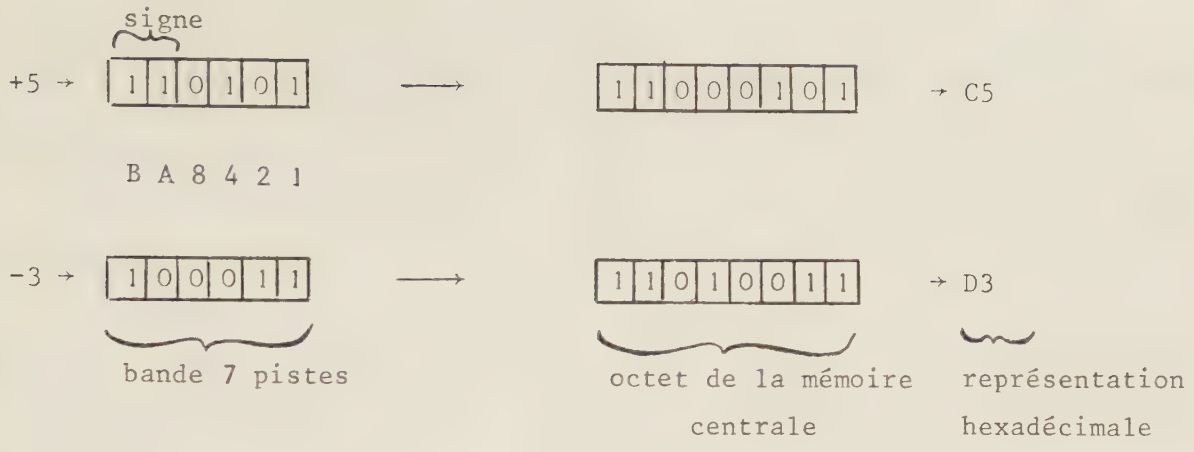
IV. - 1.2. UTILISATION DES BANDES MAGNETIQUES 7 PISTES (556 bpi).

IV.1.2.α Schéma de codage

Sur la bande enregistrée, les points de l'interférogramme sont groupés par blocs de 100 et chacun d'eux est stocké sur 6 caractères. Chaque caractère est lui-même codé sur 6 des 7 pistes de la bande, la dernière piste étant réservée pour le bit de contrôle de parité transversale. Autrement dit, chaque donnée se trouve codée avec 36 positions binaires (ou bits) sur la bande. Lorsque ces bandes sont lues sur du matériel IBM comme le 370/165 du CIRCE, chaque caractère à 6 bits est transféré de la bande vers la mémoire centrale où il est stocké dans un octet (8 bits) selon une convention appelée "7 tracks tape BCDIC". Après lecture, chaque donnée est donc codée sur 48 bits en mémoire.

Codage d'un caractère

Chaque caractère est considéré comme un chiffre décimal accompagné d'un signe. Il est codé sur 6 bits, les deux premiers définissent le signe qui est positif si ces deux bits B et A (voir schéma ci-dessous) sont égaux à 1 et négatif si seul le bit B vaut 1. La valeur binaire des quatre bits de droite sur le schéma représente le chiffre décimal en valeur absolue.



Lors de la transmission de ce caractère à 6 bits dans un octet, la convention suivante est adoptée ; les quatre bits de droite sont transférés dans les quatre bits les moins significatifs de l'octet, c'est-à-dire restent cadrés à droite, et dans les quatre autres, on stocke la représentation binaire sur 4 bits de la valeur hexadécimale C ou D suivant que le caractère est positif ((BA)) ou négatif ((B)). Le schéma ci-dessus montre sur deux exemples particuliers cette correspondance.

Codage d'une valeur de l'interférogramme

Chaque valeur est stockée avec 6 caractères sur la bande. Le premier représente la gamme qui permet de tenir compte de l'atténuation appliquée au moment de l'enregistrement de cette valeur. La gamme reste constante à l'intérieur d'un bloc de 100 valeurs, mais elle est malgré tout répétée pour chacune d'elles. Les cinq autres caractères représentent les 5 chiffres décimaux les plus significatifs de la valeur considérée avec son signe. Autrement dit, la représentation héxadécimale de la valeur -5233 (gamme 2) sera :

IV.1.2. β Lecture et décodage des valeurs

Les blocs de 100 valeurs ainsi codées sont lus par un sous-programme "PLASS". Il a été écrit en "ASSEMBLEUR 360" par J.C. CANARD puis adapté avec la collaboration de D. DRUET tous les deux ingénieurs à la compagnie IBM. Ce sous-programme se contente de transférer les blocs de valeurs de la bande vers la mémoire centrale, mais, comme on l'a vu en IV.1.1, il a l'avantage sur un

"READ" Fortran de permettre la récupération et le "DUMP" (image mémoire) automatique des blocs pour lesquels une erreur "HARDWARE" (DATA CHECK) s'est produite au cours de la lecture. Avec le "DUMP" du bloc incriminé, il donne aussi la valeur d'un certain nombre d'octets de contrôle ("SENSE BYTES") qui permettent souvent de déterminer l'origine de l'erreur.

Le décodage proprement dit de chaque caractère est fait par un sous-programme Fortran "PLECB2" écrit avec l'aide de J. COLLET qui a d'ailleurs largement contribué à la mise en route de ce système d'enregistrement sur bande magnétique 7 pistes. Ce sous-programme utilise le sous-programme "PLASS" pour la lecture et fournit en sortie, la gamme, le signe et la valeur absolue de chaque valeur. Il fait automatiquement un certain nombre de contrôles concernant la validité des blocs lus. En cas d'erreur (changement de gamme à l'intérieur d'un bloc, changement de signe dans une valeur, etc....), il sort un diagnostic en clair.

IV.1.2. ϒ Inconvénients de l'enregistrement sur 7 pistes

Le premier inconvénient de cette technique est lié à l'évolution du matériel en informatique, qui fait que les bandes 7 pistes sont progressivement abandonnées au profit des bandes 9 pistes. L'entretien du matériel enregistreur 7 pistes qui à vieilli devient donc plus difficile, et comme au niveau des lecteurs IBM, les tolérances concernant l'enregistrement physique sur la bande restent assez faibles pour des raisons de sécurité, on assiste progressivement à une augmentation du risque d'erreurs hardware au moment du décodage.

Le deuxième inconvénient, est lié à la mauvaise adaptation du codage de l'information. En dehors de la redondance caractérisée par la répétition du signe d'une valeur avec chacun de ses 6 caractères, et la répétition de la gamme pour chacune des valeurs d'un bloc, nous avons vu qu'il fallait 48 bits en mémoire centrale pour stocker une valeur qui n'était codée qu'avec 36 bits sur la bande. Au niveau du caractère proprement dit, il y a aussi une mauvaise utilisation du support d'information puisque chaque chiffre décimal d'une valeur est codé sur 4 bits et que son signe l'est sur 2 bits. Certains de ces défauts pourraient à la rigueur être justifiés par des

arguments concernant la vérification de la validité de chaque valeur. Il est certain par exemple qu'une erreur de signe sur l'un des six caractères d'un échantillon le rend suspect, mais nous avons vu qu'il est plus important d'être sûr de la position d'un point plutôt que de sa valeur. Or sur ce plan, le codage des bandes 7 pistes n'apporte aucune sécurité. En cas de point sauté, l'absence de toute numérotation des valeurs est très gênante.

Le remplacement progressif du matériel 7 pistes se fait donc au profit d'enregistreurs 9 pistes plus répandus maintenant et plus fiables. Le nouveau codage adapté pour ces bandes permet une meilleure utilisation de la capacité de stockage du support d'information et laisse une place importante à la numérotation des valeurs enregistrées.

IV.1.3. Utilisation des bandes magnétiques 9 pistes (800 bpi)

IV.1.3.α Schéma de codage

Les points de l'interférogramme sont enregistrés par blocs physiques de 170 et chacun d'eux est stocké sur trois octets soit 24 bits. La longueur de chaque bloc est donc de 510 octets. Le schéma du codage de chaque point sur trois octets est détaillé sur la figure 12 et on s'y reportera pour la numérotation des 24 bits de 0 à 23.

Le numéro du bloc où se trouve le point considéré est codé en binaire dans les quatre premiers bits (bits 0,1,2,3), seul le chiffre des unités (0 à 9) y est stocké. Il en est de même du numéro du point qui est codé dans les quatre bits suivants (bits 4,5,6,7). Le premier octet est donc entièrement et uniquement consacré au repérage de la position du point dans l'interférogramme.

Pour l'enregistrement des échantillons, on dispose d'un atténuateur à six positions qui permet de multiplier les valeurs faibles par les puissances de 4 comprises entre 1 et 1024. Les gammes correspondantes sont numérotées de 1 à 6 et codées en binaire (figure 12) sur les trois bits 8, 9 et 10 du deuxième octet.

Le signe est codé sur le bit 11. Il est négatif si ce bit est égal à 1 et positif s'il est nul.

L'intensité ou la valeur absolue du point est, elle, codée en binaire sur les douze bits qui restent et qui sont numérotés de 12 à 23. Le bit de poids le plus faible étant le $23^{ème}$. La valeur du point enregistré sera donc en valeur absolue inférieure ou égale à $2^{12}-1$ soit 4095, ce qui correspond à une valeur décimale ayant au maximum 4 chiffres significatifs.

IV.1.3.ß Lecture et décodage des valeurs ("CPLASS et HERLEC")

La lecture des blocs de 170 valeurs est faite là aussi par un sous-programme "CPLASS" en assembleur analogue au sous-programme "PLASS" de IV.1.2.ß. Comme nous l'avons vu, il remplace avantageusement l'instruction "READ" FORTRAN en cas d'erreur rencontrée à la lecture des blocs sur la bande.

Le décodage de chaque valeur est fait par un sous-programme "HERLEC" écrit en FORTRAN. La liste de ce sous-programme est donnée à la fin de ce paragraphe. Pour bien en comprendre le principe, nous allons développer l'algorithme qui en Fortran permet de trouver en décimal la valeur entière des n+1 premiers bits d'un mot de m+1 bits, lorsque la représentation des nombres entiers négatifs est faite par leur complément à 2. Cet algorithme se révèlera pratique, car s'il ne s'agit pas des n premiers bits, il est facile de les amener en tête du mot (décalage vers la gauche) en multipliant ce mot par une puissance de 2 convenable.

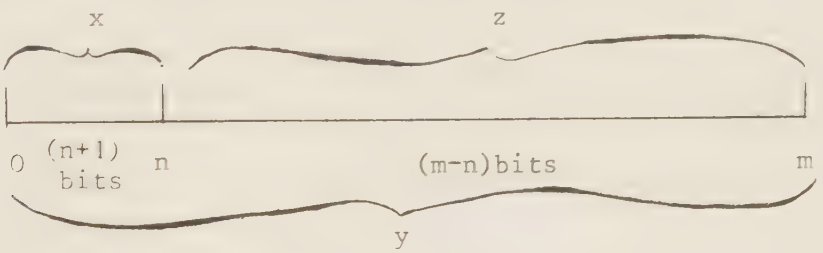
Principe du calcul de la valeur entière x des n+1 premiers bits d'un mot de m+1 bits (en Fortran).

Ce principe n'est valable que dans le cas où les nombres entiers sont représentés sur m+1 bits, le bit zéro, le plus à gauche, étant réservé au signe. (nul si le signe est positif).

Les nombres négatifs sont représentés par la méthode du complément à 2. Autrement dit, si N est un entier positif, l'entier négatif -N sera représenté par la valeur binaire de $2^{m+1}-N$. D'un point de vue pratique, pour représenter en binaire le nombre opposé à un nombre donné, on part de la droite, en se déplaçant vers la gauche, et tant qu'on trouve un zéro, on le répète pour le complément. On répète aussi le premier 1 rencontré puis on inverse chaque position logique suivante (c'est-à-dire on remplace zéro par un et inversement).

exemple : +4 est représenté par : |00000100|
-4 est représenté par : |11111100|

Soit N la valeur entière d'un mot de m+1 bits avec la représentation définie ci-dessus. Nous cherchons la valeur entière positive x correspondant à la valeur en base 2 des n+1 premiers bits de ce mot.



Soit y la valeur (entière positive) en base 2 de l'ensemble des m+1 bits (ne pas confondre avec la valeur entière N positive ou négative représentée par ce mot) et z la valeur en base 2 des (m-n) derniers bits. Nous avons la relation :

(140) $\quad y = x.2^{m-n} + z \qquad$ avec $z < 2^{m-n}$

Pour trouver la valeur de x en fonction de N, nous devons envisager quatre cas différents :

a) $N \neq 0$

Dans ce cas, le bit de signe est nul et nous avons directement :

$$N = y = x.2^{m-n} + z$$

Comme x est entier et comme $z < 2^{m-n}$, nous obtenons :

(141) $\quad \boxed{x = E\left[\frac{N}{2^{m-n}}\right]} \qquad$ où : $E[x]$ = partie entière de x

b) $N < 0$ et $N = -k.2^{m-n}$ avec k entier > 0

Lorsque N est négatif, y est le complément à 2 de la valeur binaire de -N d'où :

$$y = x.2^{m-n} + z = 2^{m+1} + N$$

qui s'écrit :

(142) $$x.2^{m-n} = 2^{m+1} + N - z$$

La valeur entière $-N = k.2^{m-n}$ correspond à un mot où les bits de n+1 à m sont nuls. Il en sera de même de son complément à 2 qui correspond à la valeur entière N négative. On en déduit donc : $\boxed{z = 0}$. C'est un des moyens de reconnaître la condition b. En portant dans (142), nous avons :

$$x.2^{m-n} = 2^{m+1} - k.2^{m-n}$$

$$x = 2^{n+1} - k$$

D'où :

(143) $$\boxed{x = 2^{n+1} + E\left[\frac{N}{2^{m-n}}\right]} \Longrightarrow z = 0$$

c) $N < 0$ et $N = -k.2^{m-n} - z'$ avec $\begin{cases} k \text{ entier} > 0 \\ z' \text{ entier} > 0 \text{ infér. à } 2^{m-n} \end{cases}$

Portons la valeur de N dans (142) :

$$x.2^{m-n} = 2^{m+1} - k.2^{m-n} - z' - z$$

$$x = 2^{n+1} - k - \frac{z+z'}{2^{m-n}}$$

Etant donné que $z' \neq 0$ et que, par hypothèse, $(z + z') < 2^{m+1-n} - 1$, la seule possibilité pour que x soit entier est que la quantité $(z + z')$ soit égale à 2^{m-n}. D'où l'expression :

(144) $$\boxed{x = 2^{n+1} + E\left[\frac{N}{2^{m-n}}\right] - 1}$$

Vérifions sur un exemple que la quantité (z+z') vaut bien 2^{m-n}.

EXEMPLE :

$N = 2^{m-n} + (2^{m-n} - 1) \rightarrow k = 1$ et $z' = 2^{m-n} - 1 \rightarrow z = 1$

$N \rightarrow |0 \text{——} 01|1 \text{————————} 1|$ $\quad -N \rightarrow |1 \text{——} 10|0 \text{————————} 01|$

0 n m 0 n m

$N = 2.2^{m-n} \rightarrow k = 2$ et $z' = z = 0$ (cas particulier vu en b)

$N \rightarrow |0 \text{——} 010|0 \text{————————} 0|$ $\quad -N \rightarrow |1 \text{——} 10|0 \text{————————} 0|$

0 n m 0 n m

$N = 2.2^{m-n} + 1 \rightarrow k = 2$ et $z' = 1 \rightarrow z = 2^{m-n} - 1$

$N \rightarrow |0 \text{——} 010|0 \text{————————} 01|$ $\quad -N \rightarrow |1 \text{——} 101|1 \text{————————} 1|$

0 n m 0 n m

$N = 2.2^{m-n} + 2 \rightarrow k = 2$ et $z' = 2 \rightarrow z = 2^{m-n} + 2$

$N \rightarrow |0 \text{——} 010|0 \text{————————} 010|$ $\quad -N \rightarrow |1 \text{——} 101|1 \text{————————} 110|$

0 n m 0 n m

---

d) $N < 0$ mais $|N| < 2^{m-n}$ ou $N = -z'$ avec $z' < 2^{m-n}$

Il s'agit en fait d'un cas particulier de c avec k = 0. Cette fois ci, nous aurons :

$$x.2^{m-n} = 2^{m+1} + N - z = 2^{m+1} - z' - z$$

$$x = 2^{n+1} - \frac{z'+z}{2^{m-n}}$$

Là aussi on aura toujours $z+z' = 2^{m-n}$ d'où :

(145) $\boxed{x = 2^{n+1} - 1}$

Cette formule, implique que tous les bits de x sont égaux à 1, ce qu'on peut vérifier sur l'exemple ci-dessous :

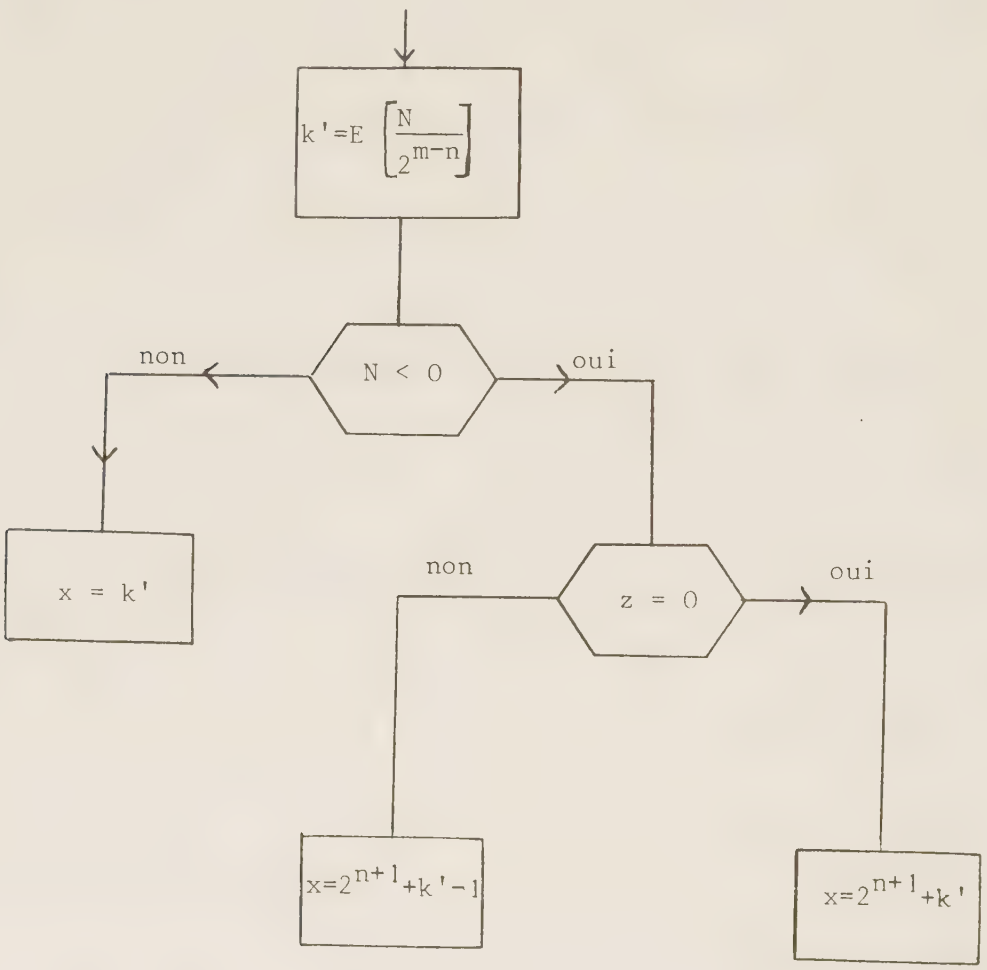
EXEMPLE :

$N = 2^{m-n} \rightarrow$ | 0 ... 01 | 0 ... 0 | et $-N \rightarrow$ | 1 ... 1 | 0 ... 0 |

$N = 2^{m-n} - 1 \rightarrow$ | 0 ... 0 | 1 ... 1 | et $-N \rightarrow$ | 1 ... 1 | 0 ... 01 |

$N = 1$ | 0 ... 0 | 0 ... 01 | et $-N \rightarrow$ | 1 ... 1 | 1 ... 1 |

ORGANIGRAMME DU CALCUL DE x



- calcul de la valeur entière x des n+1 premiers bits d'un mot de m+1 bits en Fortran. N est l'entier ≥ 0 ou < 0 représenté par ce mot.

Application au sous-programme "HERLEC"

Sur le 370/165 IBM, nous avons m+1 = 32 et les groupes de trois octets sont lus en format A3 et cadrés à gauche dans les mots de quatre octets. Le dernier octet de droite est rempli avec le caractère blanc (40 en hexadécimal).

Dans une première étape, on cherche la valeur x des 8 premiers bits (n=7). On calcule donc $k'=N/2^{24}$ et suivant que N la valeur entière du mot est positive ou négative, on calcule :

$$\begin{cases} x = k' \quad si \quad N \geq 0 \\ x = k'-1 + 2^8 = k' + 255 \quad si \quad N < 0 \end{cases}$$

Le cas b où z = 0 est à exclure puisque le dernier octet n'est pas nul.

A partir de x il est facile de calculer le numéro du bloc et le numéro de la mesure.

Pour calculer la valeur des 3 bits de gamme, il suffit d'amener ces trois bits sur la frontière gauche de ce mot en le multipliant par 2^8. L'algorithme général du calcul de x peut être appliqué à ces 3 bits. Si on multiplie le mot par 2^3, c'est le bit de signe de la valeur à décoder qui se trouve sur la frontière gauche du mot avec la même convention que celui des nombres entiers en général. La valeur de la représentation binaire des 12 bits d'intensité et du bit de signe est calculée avec l'algorithme déjà utilisé pour la numérotation et la gamme. Si le signe est négatif, on annule la contribution du bit correspondant en enlevant 2^{12} à la valeur ainsi trouvée.

Dans le programme, c'est "K" ou "NMES(J)" qui joue le rôle de N.

IV.1.3. ϒ Avantages de l'enregistrement sur 9 pistes

Cette solution qui de toute façon se serait imposée du fait de l'évolution du matériel informatique s'est révélée plus fiable sur le plan technique. Elle permet de stocker plus d'informations sur une même longueur de bande Cette dernière amélioration est due à plusieurs facteurs : la densité plus élevée ; le nombre de valeurs par bloc qui passe de 100 à 170 ; une meilleure utilisation du support d'information grâce au codage bien adapté à l'ordinateur utilisé. La numérotation des blocs et des mesures bien que partielle s'est avérée très utile et nous pensons dans un proche avenir affecter un octet complet pour chacune de ces deux numérotations pour augmenter la sécurité.

Nous avons aussi obtenu un gain appréciable sur la durée du décodage. En effet, le temps mis pour lire, décoder, puis recopier 10 000 valeurs (en binaire sur une autre bande 9 pistes, 1600 bpi) est passé de 0,88 secondes avec le système 7 pistes à 0,30 secondes avec le système 9 pistes. Une amélioration qui reste à essayer est de remplacer le sous-programme "HERLEC" par un autre, équivalent, mais écrit en langage "assembleur" mieux adapté pour le décodage proprement dit.

IV.1.3.δ Liste du sous-programme "HERLEC"

```
      SUBROUTINE HERLEC(D,IMP,N170,IEOF)
C
C     SOUS PROGRAMME DE LECTURE ET DECODAGE DES BANDES 9 PISTES (800 BPI.)
C   LA LECTURE DE CHAQUE BLOC PEUT ETRE FAITE SOIT PAR UN READ FORTRAN (LECFORT)
C   SOIT PAR L'APPEL AU SOUS PROGR. ''CPLASS'' (PLASS).
      DIMENSION D(170),IG(7),NMES(170),NBLOC(170)
      IG(1)=1024
      IG(2)=256
      IG(3)=64
      IG(4)=16
      IG(5)=4
      IG(6)=1
      IG(7)=0
      IERG=0
C*************************************************************************
C  LECTURE D'UN BLOC DE NS MESURES DE 3 OCTETS (ICI NS=170)              *
C                                                                        *
C/GO.FT13F001 DD UNIT=BDE9,DISP=OLD,LABEL=(1,NL),DSN=FT13101,          LECFORT*
C/ DCB=(RECFM=F,BLKSIZE=510,LRECL=510),VOL=SER=9127                    LECFORT*
C     READ(13,130,END=888)NMES                                         LECFORT*
C 130 FORMAT(170A3)                                                    LECFORT*
C*************************************************************************
      KVRET=0                                                          PLASS
      CALL CPLASS(NMES,KVRET)                                          PLASS
      IF(KVRET.EQ.2)GO TO 888                                          PLASS
      IF(KVRET.EQ.1)PRINT 505,N170                                     PLASS
  505 FORMAT(//,' ERREUR DETECTEE PAR PLASS DANS LE BLOC',I10,//)      PLASS
C                                                                      PLASS
      IF(IMP.EQ.2)PRINT 500,NMES
  500 FORMAT(1X,12Z10)
C
C  CALCUL DE LA VALEUR DECIMALE POSITIVE DE LA REPRESENTATION BINAIRE
C  DU PREMIER OCTET SOIT NBB.
      DO 1 J=1,170
      K=NMES(J)
      NBB=K/16777216
      IF(K.LT.0)NBB=NBB+255
C
C  CALCUL DE NMES(I),NBLOC(I) DANS LE PREMIER OCTET DE CHAQUE MES.DU BLOC.
      NBLOC(J)=NBB/16
      NMES(J)=NBB-NBLOC(J)*16
C
C    CALCUL DE LA GAMME DANS LES 3 PREMIERS BITS DU DEUXIEME OCTET.
      K=K*256
      IGAM=K/536870912
      IF(K.LT.0)IGAM=IGAM+7
      IF(IGAM.NE.0.AND.IGAM.LT.7)GO TO 55
C *****CAS OU IL Y A UNE ERREUR DE GAMME.
      IERG=IERG+1
      IGAM=7
C
C   CALCUL DU SIGNE ET DE L'INTENSITE
   55 K=K*8
      IF(K.GE.0)GO TO 22
C     D(J)=-(8191+K/524288-4096)*IG(IGAM)
      D(J)=(-4095-K/524288)*IG(IGAM)
      GO TO 1
   22 D(J)=(K/524288)*IG(IGAM)
    1 CONTINUE
C
      IF(IERG.EQ.0)GO TO 57
      PRINT 56,IERG,N170
   56 FORMAT(//,'  IL Y A ',I4,' ERREURS DE GAMME DANS LE BLOC',I10,//)
   57 IF(IMP+IERG.GT.0)PRINT 54,(NBLOC(L),NMES(L),D(L),L=1,170)
   54 FORMAT(/,(1X,10(I2,I2,F9.0)))
C
      GO TO 886
  888 MEOF=(N170-1)*170
      PRINT 887,N170,MEOF
  887 FORMAT(//,' E.O.F. TROUVEE EN VOULANT LIRE LE BLOC',I8,' M=',I8,/)
      IEOF=9999
  886 RETURN
      END
```

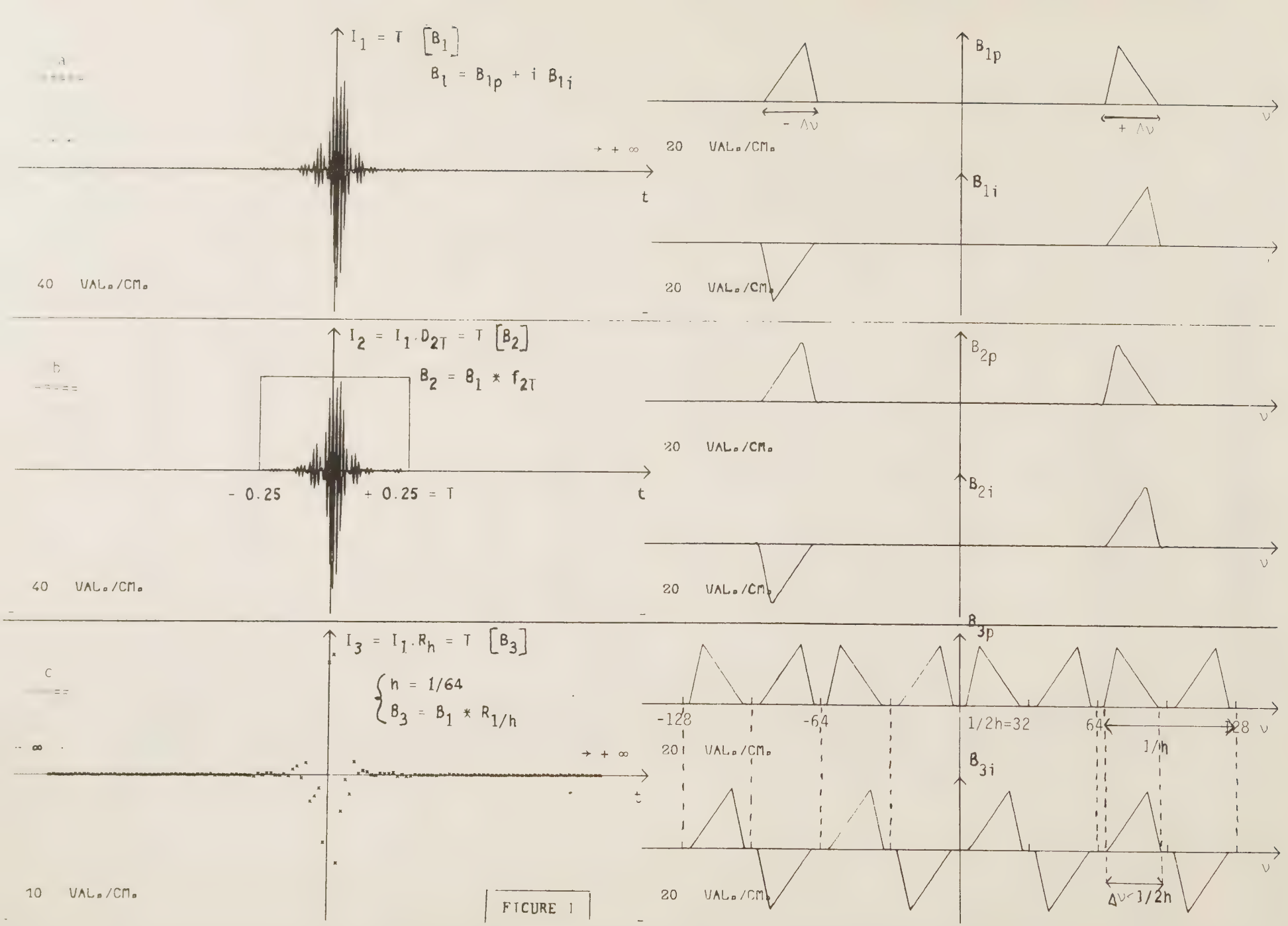

a
$I_1 = T\ [B_1]$
$B_l = B_{1p} + i\ B_{1i}$
$\rightarrow +\infty$
t
40 VAL./CM.
B_{1p}
$-\Delta\nu$
$+\Delta\nu$
20 VAL./CM.
B_{1i}
20 VAL./CM.
b
$I_2 = I_1 . D_{2T} = T\ [B_2]$
$B_2 = B_1 * f_{2T}$
- 0.25
+ 0.25 = T
t
40 VAL./CM.
B_{2p}
20 VAL./CM.
B_{2i}
20 VAL./CM.
c
$I_3 = I_1 . R_h = T\ [B_3]$
$h = 1/64$
$B_3 = B_1 * R_{1/h}$
$-\infty$
$\rightarrow +\infty$
t
10 VAL./CM.
B_{3p}
-128
-64
1/2h=32
64
128 ν
1/h
20 VAL./CM.
B_{3i}
$\Delta\nu = 1/2h$
20 VAL./CM.

FIGURE 1

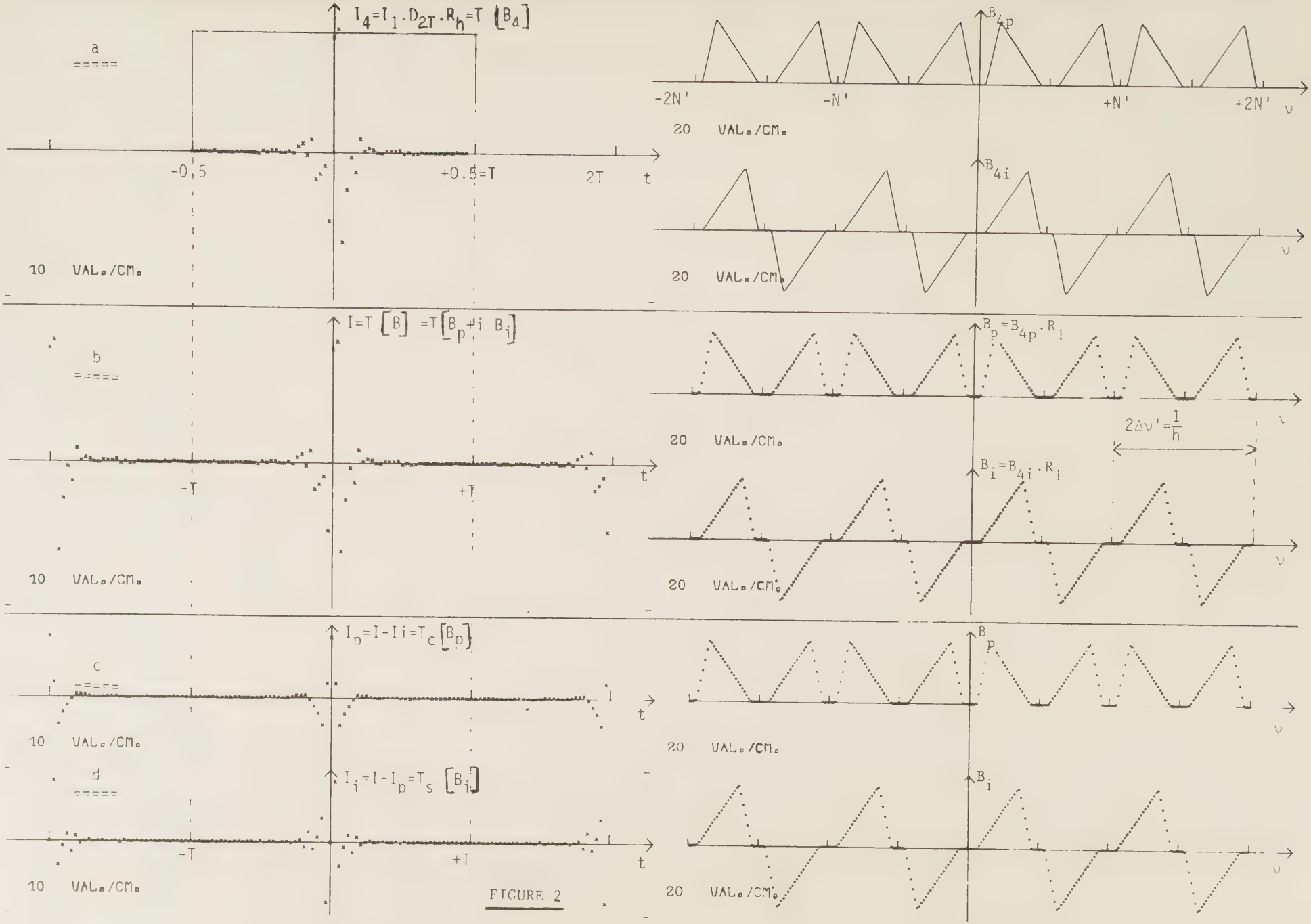
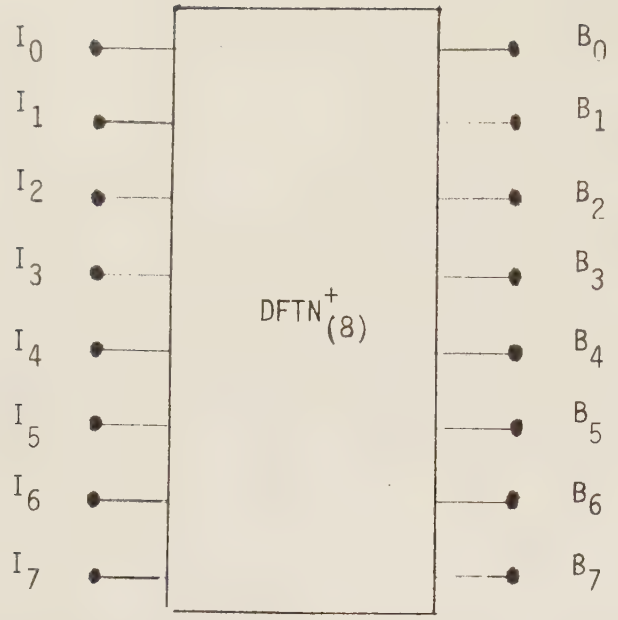
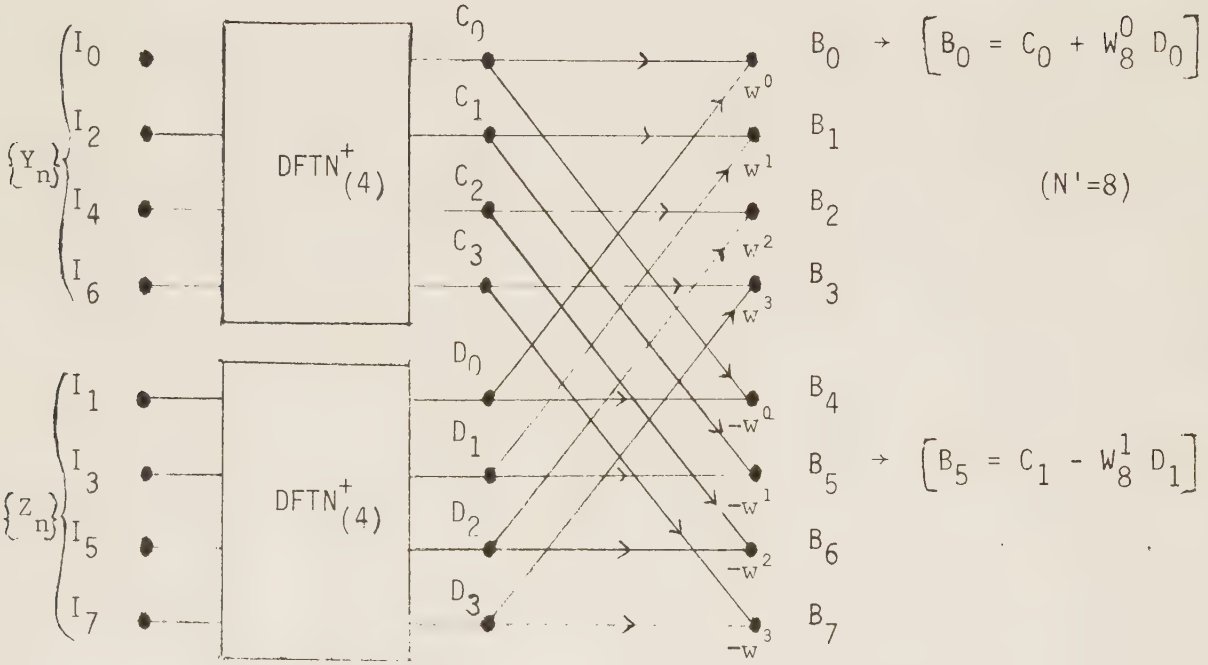

a
$I_4=I_1 \cdot D_{2T} \cdot R_h=T\ [B_4]$
-0,5
+0.5=T
2T
t
10 VALo/CMo
B_{4p}
-2N'
-N'
+N'
+2N'
ν
20 VALo/CMo
B_{4i}
20 VALo/CMo
b
$I=T\ [B] = T[B_p + i\ B_i]$
-T
+T
10 VALo/CMo
$B_p=B_{4p} \cdot R_l$
20 VALo/CMo
$2\Delta\nu'=\frac{1}{h}$
$B_i=B_{4i} \cdot R_l$
20 VALo/CMo
c
$I_p=I-Ii=T_c\ [B_p]$
10 VALo/CMo
B_p
20 VALo/CMo
d
$I_i=I-I_p=T_s\ [B_i]$
10 VALo/CMo
B_i
20 VALo/CMo

FIGURE 2



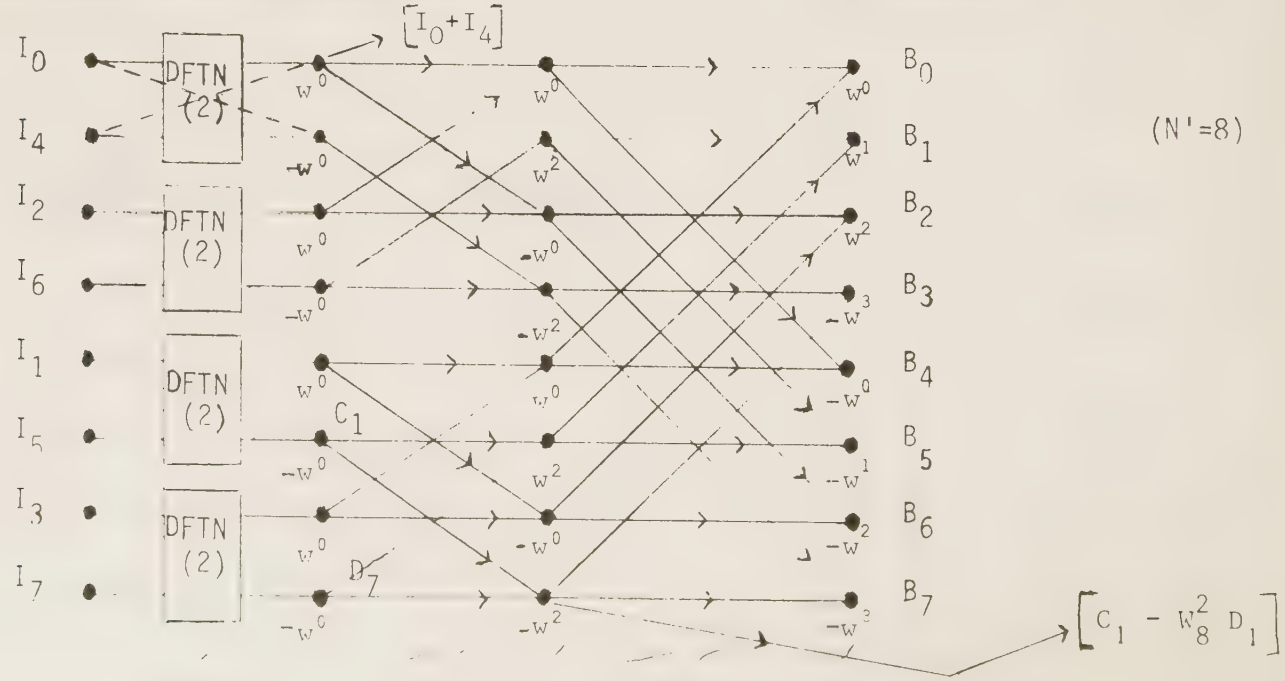
(N'=8)

FIGURE 3

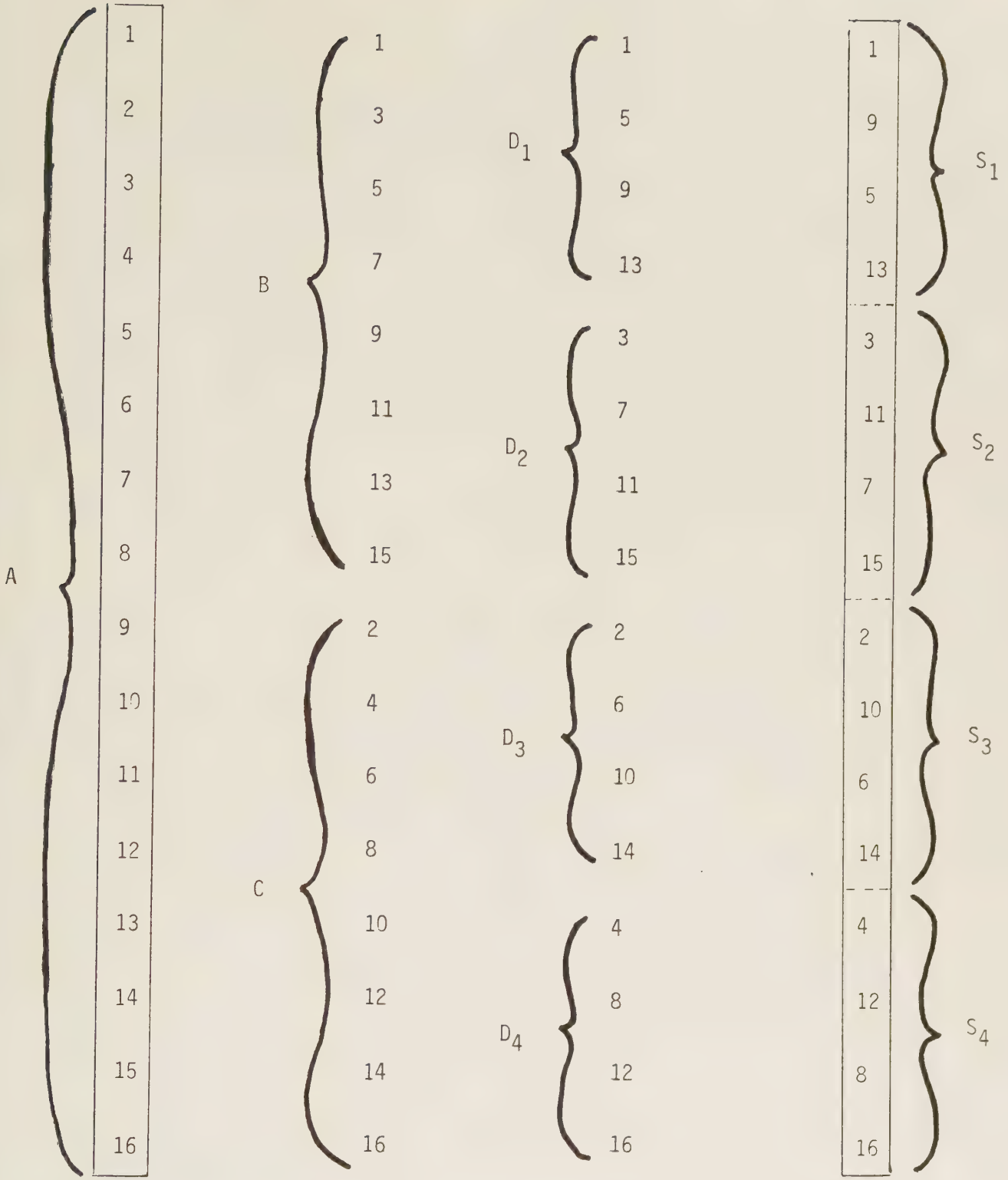


(N'=8)

FIGURE 4



(N'=8)



- Indices dans l'ordre naturel (série initiale A)

- Indices ordonnés suivant la règle dite de l'inversion de bits.

FIGURE 6 - Remplacement du tri "par inversion de bits" par une décomposition progressive de chaque série en deux séries dont l'une contient les indices de rang pair et l'autre ceux de rang impair.

PROGRAMME	DFT classique	FFT "JPAUL"	FFT32T V1 (B.I.)	FFT32T V3	FFT64K V3	RHARM IBM	FFT128 V2	FFT128 V3	FFT1024K avec disq	TFID4 avec disq	T101	
FONCTION	complexe	complexe	complexe	complexe	réelle	réelle	réelle impaire	réelle impaire	réelle	réelle impaire		
PLACE en MEMOIRE CENT	8N	4N	5N	5N	5N/2	5N/2	5N/4	5N/4	$\frac{5Nmax}{32}$	15K →80K (48K)		
N'=2N=1K	5 000	20		6	4	5		2				← 370/165
		53	34	16	10	12	12	6				← 360/75
N'=2K	20 000	47		17	9	10		4				
		124	71	38	21	25	23	12				
N'=4K	13'	116		40	23	20		10				
		270	148	82	47	54	47	25				
N'=8K	52'	264		90	47	46		23				
		596	312	178	98	118	100	58				
N'=16K	4h	602		218	104	98		50				
		1 294	644	382	208	244	210	115				
N'=32K		1 373		484	234	205		113				
		2 812	1 340	820	440	510	438	245				
N'=64K		2 912		1 028	533	436		278		525	213	
		6 106	2 788		955	1 070	908	530	4 292			
N'=128K					1 132			572		1 200	344	
							1 956	1 090	8 400			
N'=256K								1 240		2 400	550	
									16 600			
N'=512K										50"	10"	
									5'27"			
N'=1024K										1'51"	17"	
									11'			
N'=2048K										3'57"	34"	
									22'08"			
N'=4096K										8'16"	1'10"	

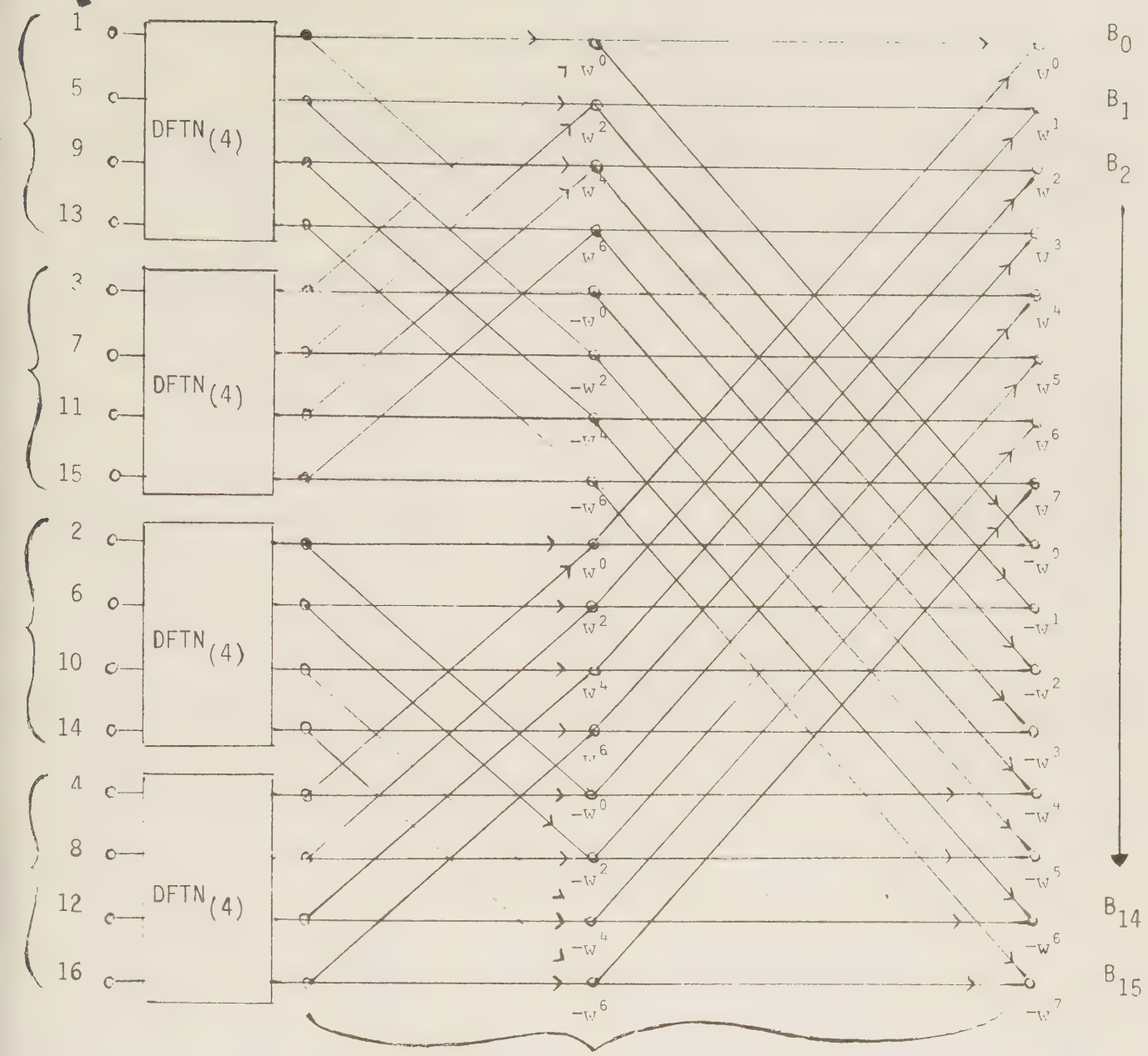
FIGURE 7

Tableau récapitulatif des temps de calcul sur 370/165 et 360/75 IBM.

- En l'absence de toute indication, les durées sont exprimées en 1/100ème sec. et sinon en minutes (') et secondes (")
- N' est le nombre d'échantillons de la fonction enregistrée de -T à +T. Dans le cas d'une fonction réelle impaire, seuls les N=N'/ échantillons enregistrés de 0 à +T sont utilisés.
- Pour chaque valeur de N' il y a deux indications de durée, la première correspond au 370/165 et la deuxième au 360/75.
- La version V1 des programmes utilise "l'inversion de bits" (faite par un sous-programme en assembleur). La version V2 ne l'utilis pas mais n'apporte qu'un gain très faible sur la durée (<1/10). La version V3 est une optimisation en ce qui concerne la durée su

I_n

ries n+1



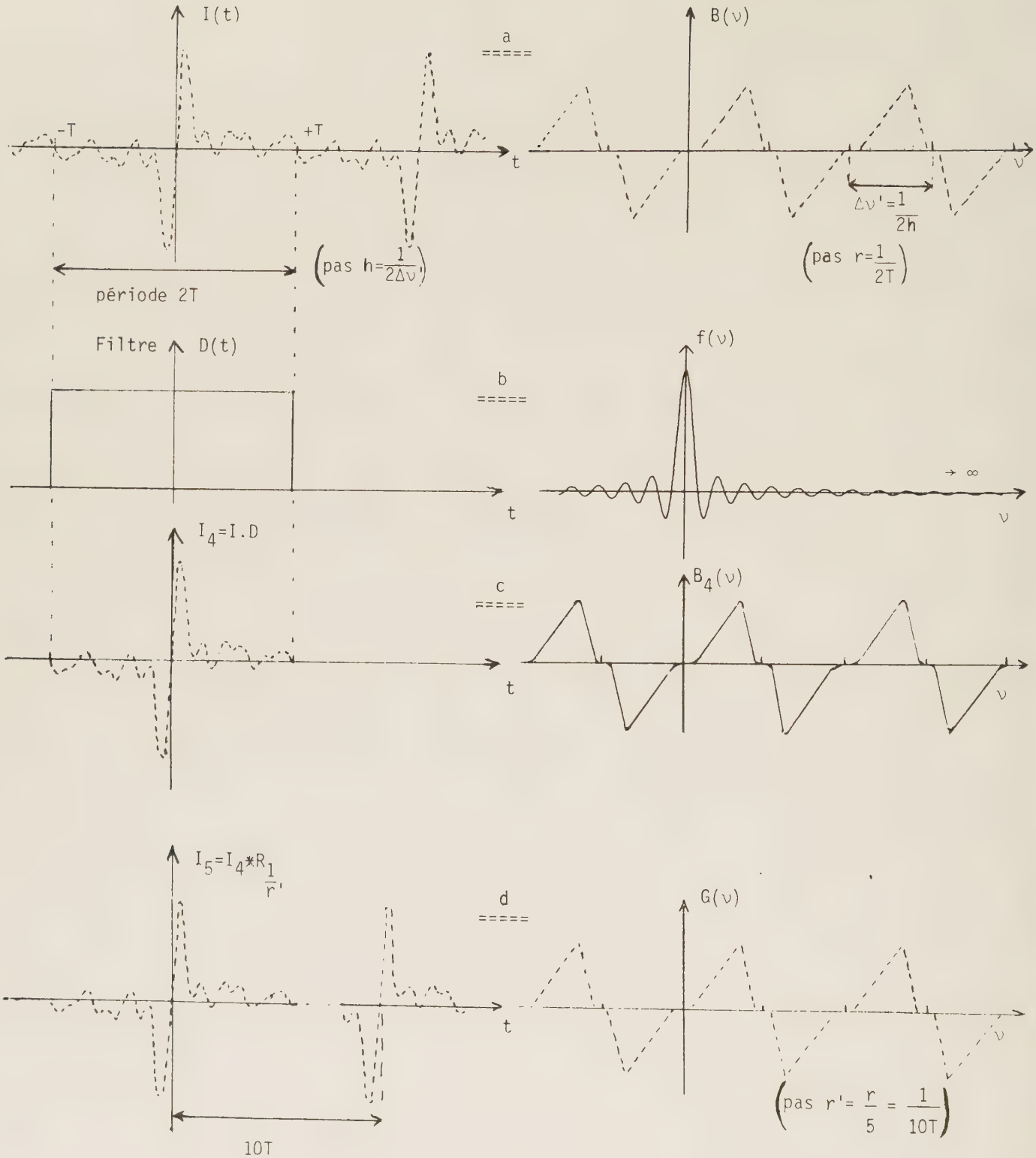
p. étapes d'intercombinaisons

(N=16, n_b=4, p=2, N_T=4)

$$\{I_n\}_N \xrightarrow{DFTN_{(N)}} \{B_k\}$$

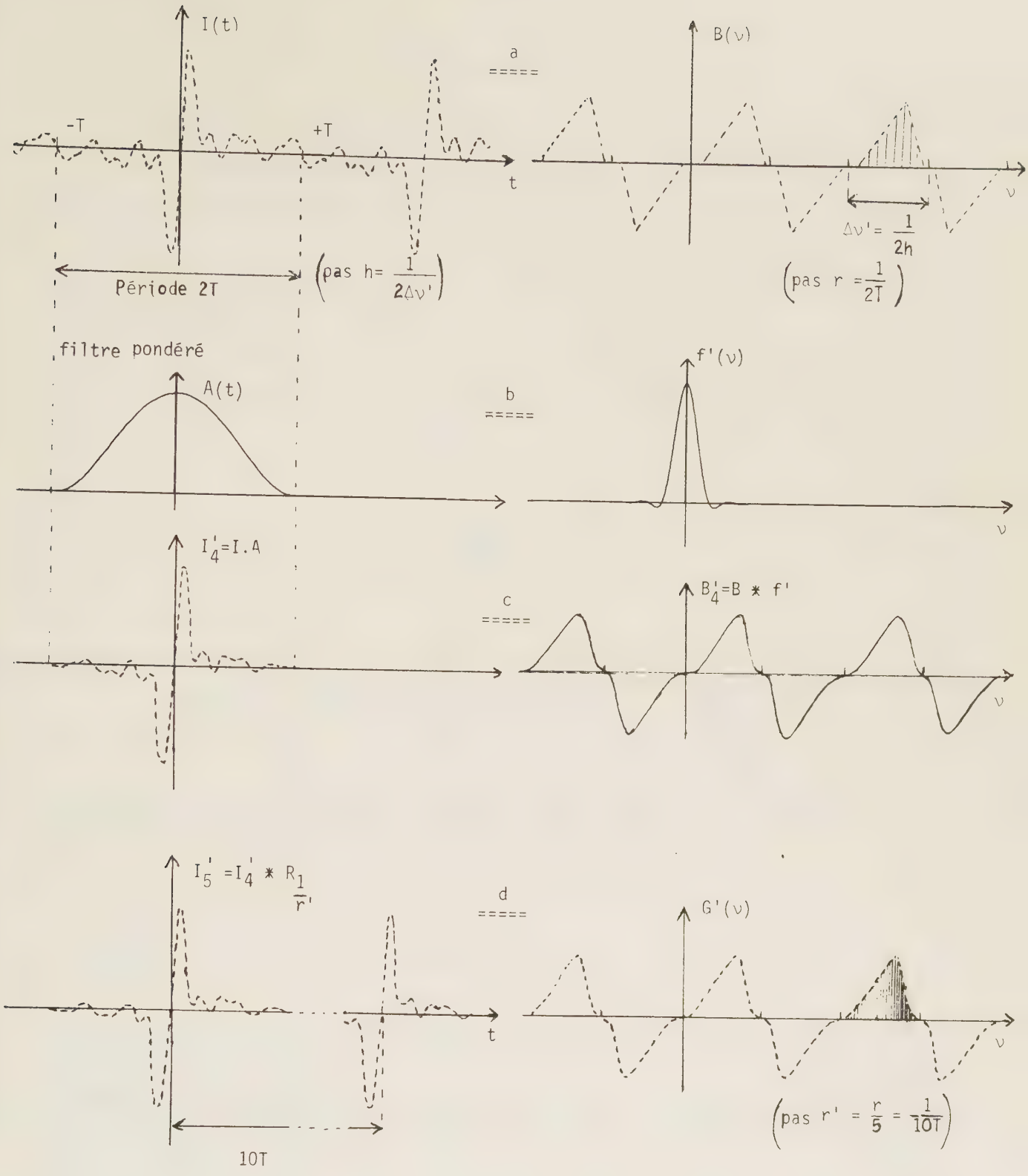
$$\{I_n\}_N \xrightarrow{\text{Tri}} \left\{ \begin{matrix} n_b \\ \text{séries} \\ D_j \end{matrix} \right\} \xrightarrow{n_b \times DFTN_{(N/n_b)}} \left\{ \begin{matrix} n_b \\ \text{séries} \\ \text{transfor-} \\ \text{mées} \end{matrix} \right\} \xrightarrow{p \text{ intercombinaisons}} \{B_k\}_N$$

FIGURE 8 : Principe de base permettant la décomposition d'une $DFTN_{(N)}$ en n_b $DFTN_{(N/n_b)}$.



INTERPOLATION DANS LE SPECTRE SANS APODISATION

FIGURE 9



INTERPOLATION DANS LE SPECTRE AVEC APODISATION

FIGURE 10

- Fonctions d'appareil pour différentes valeurs du paramètre "IAP".

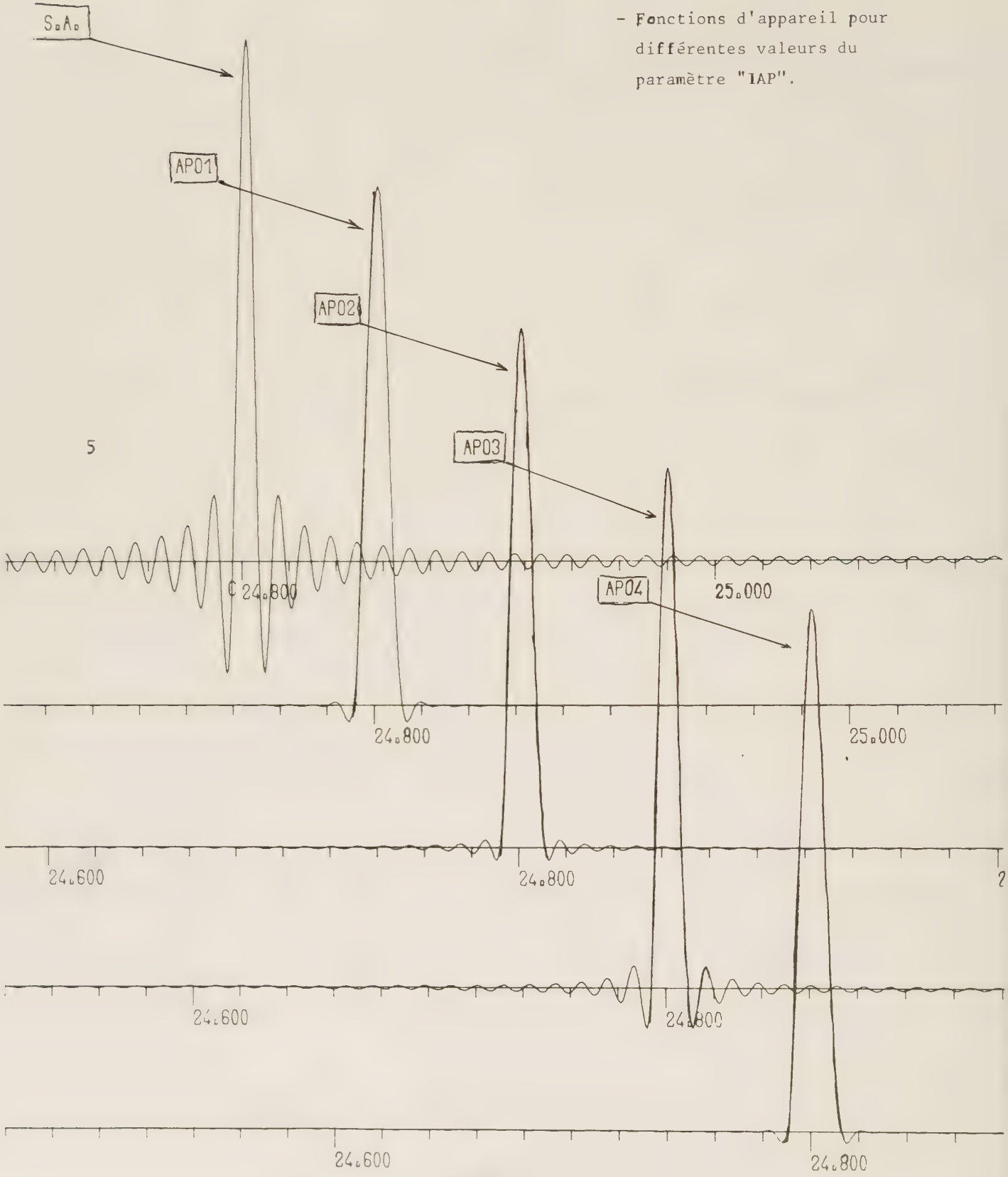
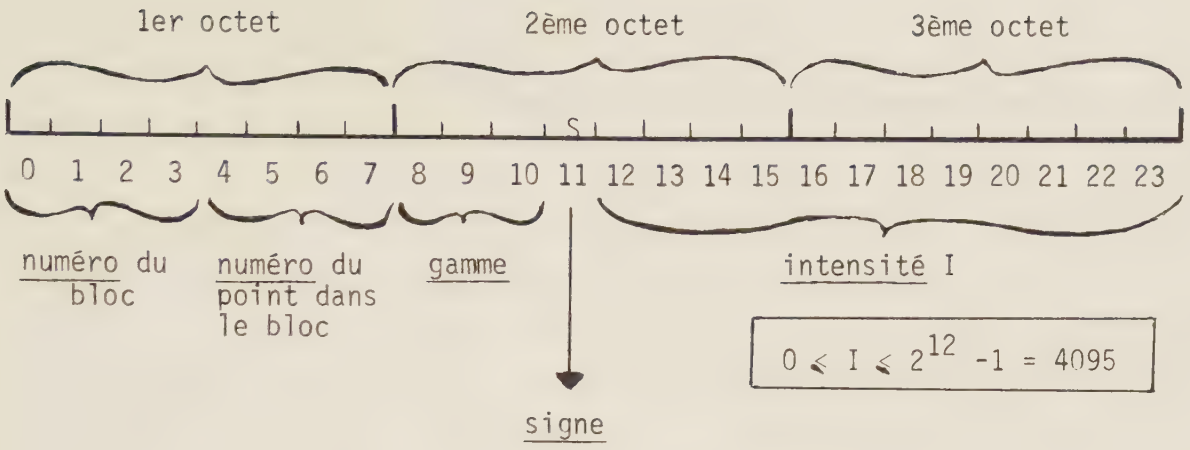


FIGURE 11



- Les blocs et les points sont numérotés cycliquement sur quatre bits de la façon suivante : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2 etc.....

- Signe : { négatif si le bit 11 est égal à 1
  positif si le bit 11 est nul

- Intensité : Valeur de la représentation binaire des bits 12 à 23, le bit 12 étant celui de plus fort poids.

- Gamme :

Valeur de la gamme	Gain à l'enregistrement	Gain à la lecture	Bits		
			8	9	10
1	1	1024	0	0	1
2	4	256	0	1	0
3	16	64	0	1	1
4	64	16	1	0	0
5	256	4	1	0	1
6	1024	1	1	1	0

FIGURE 12 - Le codage de la valeur d'un point de l'interférogramme sur une bande 9 pistes.

CONCLUSION

Le calcul de la T.F. de l'interférogramme a cessé d'être un handicap pour la spectroscopie par transformée de fourier.

Les progrès d'ordres techniques (hardware et software) réalisés en informatique, la découverte de l'algorithme F.F.T. et les améliorations successives apportées à sa programmation et à son adaptation à la transformation d'une fonction réelle impaire, font que la durée de ce calcul ne dépasse pas le dixième de celle qui serait de toute façon nécessaire à l'exploitation proprement dite d'un spectre de plus de un million d'éléments spectraux. En effet, l'enregistrement, le stockage, le tracé, les corrections (augmentation du rapport s/b par des moyennes par exemple), et la détermination précise de la position et de l'intensité des raies de tels spectres ne peuvent être envisagés qu'avec des méthodes numériques qui sont identiques quelle que soit la méthode de spectroscopie utilisée pour l'obtention du spectre.

D'autre part, avec le programme "TFID4" , la transformation d'un grand nombre de valeurs reste possible sur un ordinateur ayant peu de place en mémoire centrale à condition qu'on puisse disposer d'une unité de disque magnétique ou son équivalent comme support auxiliaire de mémoire.

De nombreux chercheurs travaillant dans ce domaine et qui n'avaient pas de connaissances approfondies en informatique ont été amenés à se servir de ces programmes pour transformer et exploiter leurs interférogrammes. Avec l'aide des deux brochures détaillant le mode d'emploi de ces programmes, ils ont pu mener à bien leurs calculs sans que le temps passé à leur préparation puisse être considéré comme un handicap pour leurs travaux de recherche. De plus, le fait que ces programmes soient presque tous écrits en fortran, favorise leur compréhension et leur éventuelle adaptation par le scientifique qui le désire.

Sur le plan de la précision de l'ensemble des calculs, l'évaluation de l'incertitude introduite sur la mesure de la position d'une raie constitue un test probant pour le spectroscopiste. En partant d'un interférogramme synthétique dont l'échantillonnage peut être calculé avec une incertitude négligeable, nous avons calculé son spectre avec la chaîne des programmes habituellement utilisés et l'erreur relative introduite par les calculs est inférieure à 10^{-9} . A partir d'un interférogramme réel, nous avons montré [38] [39] que l'incertitude relative se situait vers $2{,}5.10^{-8}$, ce qui est légèrement supérieur à l'ordre de grandeur des erreurs systématiques prévisibles sur le même résultat. Pour que les erreurs de calcul restent négligeables, nous serons peut être amenés à l'avenir (pour certaines mesures précises) à augmenter la précision de certaines parties des programmes et en particulier de l'interpolation qui est facilement perfectible sur ce plan aux dépens de la durée. Les calculs tels qu'ils sont donnent satisfaction mais il est certain que toute amélioration dans le domaine des précisions exigerait une recherche systématique des raisons qui la limitent actuellement à cette valeur.

En dehors de cette étude assez délicate, les projets d'avenir dans le domaine des programmes portent :

- sur la mise au point d'un nouveau programme "TEST 8" qui ne fasse plus appel à un filtrage numérique pour calculer rapidement une tranche du spectre, mais à une technique décrite sous le nom de "F.F.T. Pruning" par John D. MARKEL [40] .

- l'écriture de programmes analogues à "FFT128" et "TFID4" mais adaptés au cas des fonctions paires pour le premier et réelles paires ou non pour le second. En effet, les programmes d'intérêt général ("FFT32T", "FFT64K", "FFT128", et "TFID4"), déjà mis au point sont utilisés par de nombreux chercheurs travaillant dans des domaines d'application très variés, et pour pouvoir mieux s'adapter à chaque cas particulier, il est nécessaire de compléter la gamme de ces programmes "spécialisés" pour un type particulier de fonctions.

- l'écriture du sous programme de décodage "HERLEC" en "assembleur" pour essayer de diminuer le temps de lecture des bandes enregistrées.

Je tiens, en terminant, à exprimer à Madame J. CONNES qui a dirigé ce travail, l'expression de ma profonde reconnaissance pour l'intérêt constant, les précieux conseils et la sollicitude qu'elle a bien voulu me témoigner.

Je remercie Monsieur le Professeur JACQUINOT, Directeur du Laboratoire Aimé Cotton, d'avoir bien voulu me faire l'honneur de présider ce jury.

Je prie Messieurs les Professeurs J. ARSAC et M. FRANÇON d'accepter l'expression de ma respectueuse gratitude pour avoir bien voulu faire partie du jury en dépit de leurs nombreuses occupations.

Je remercie également, tout le personnel du CIRCE qui m'a aidé dans la mise au point des programmes et l'édition des brochures explicatives, et tout spécialement Madame F. GAUTIER pour sa collaboration au sein de notre équipe et Madame M. GARNIAUX qui a assuré la publication de ce travail dans un temps record !

Je suis heureux d'exprimer toute ma reconnaissance aux chercheurs et techniciens du Laboratoire Aimé Cotton parmi lesquels, P. CONNES, J. BLAISE. G. GUELACHVILI, J.P. MAILLARD, J. PINARD, G. MICHEL, J. CHEVILLARD, C.AMIOT, R. VETTER, avec qui j'ai toujours eu de très sympathiques et très fructueux contacts.

REFERENCES

1 MAILLARD (J.P.) - Thèse, Fac. des Sciences d'Orsay, 1967.

2 CONNES (J.) et (P.), DELOUIS (H.), GUELACHVILI (G.), MAILLARD (J.P.), MICHEL (G.) - Nouv. Rev. d'Opt. Appliq. 1970, 1, n° 1, p. 3-22.

3 CONNES (J.) - Thèse, Revue d'Optique, 1961.

4 CONNES (J.) et NOZAL (V.) - Jl. Phys. Juin 1961 p. 359.

5 CONNES (J.) et (P.) - J.O.S.A., 1966, 56, 896.

6 CONNES (J.) et (P.), MAILLARD (J.P.) - Jl Phys. 1967, 28, C2, 120

7 DELOUIS (H.) - Thèse, Fac. des Sciences d'Orsay, 1968.

8 PINARD (J.) - Jl. Phys., 1967, 28, C2, 136.

9 PINARD (J.) - Thèse, Ann. Phys., 1969, n° 2.

10 SHANNON (C.):Bell Syst. Tech. - Jl 1948, T.3 p. 47 et 379.

11 SHANNON (C.E.) : Proc. IRE, 37, 10 (1949).

12 SHANNON (C.), WEAVER (W.) : The mathematical theory of communication, Univ. Illinois Press., p. 53 (1949).

13 KOHLENBERG (A.) Jl. Appl. Physics, t. 24, 1953, p. 1432.

14 ZAKAI (M.), Information and Control, 1965, t.8, p. 143-158.

15 CAMPBELL (L.L.), SIAM (J.), App. Math., 1968, t.15, p. 626.

16 CONNES (J.) et (P.) - Table ronde du J.P.L. - Avril 1964.

17 COOLEY (J.W.) et TUCKEY (J.W.) - Mathématics of computation, 1965, 19, 296.

18 GOOD (I.J.) - J. Roy. Statist. Ser. B, vol. 20, p. 361-372, 1958.

19 COOLEY (J.W.), LEWIS (P.A.W.), WELCH (P.D.) - IEEE transac. Aud. El. Acous., june 1967, vol. Au-15, NO.2 p. 76.

20 BINGHAM (C.), GODFREY (M.D.), TUCKEY (J.W.), même publication p. 56-66.

21 COOLEY (J.W.) - Proc. of the IBM Scient. Comput. Symp. Juin 1966.

22 BRENNER (N.M.) - Communication privée (sept. 1968).

23 COOLEY (J.W.), LEWIS (P.A.W.), WELCH (P.D.) - IBM research RC 1743 (1967).

24 BRIGHAM (E.O.), MORROW (R.E.) - IEEE spectrum, Décembre 1967.

25 CLIFFORD (J.) WEINSTEIN - IEEE, trans. aud. Elec., vol.Au-17, n°3, sept. 1969.

26 BERGLAND (G.D.) - Bell Tel. Laborat., Incorporated - Whippany, New Jersey - 1967 (p. 236-238).

27 BUIJS (H.L.) - Applied optics 211, vol. 8, n° 1, 1969.

28 GOOD (I.J.) - IEEE transac. on comput.- Mars 1971 - p. 310.

29 RAMOS (G.U. - Mathém. of comput., vol. 25, n° 116 - p. 757. octobre 1971.

30 SINGLETON (R.C.) - Communications of the A.C.M. - Vol. 10, n° 10 - octobre 1967 (p.647) et vol. 11, n° 11, novembre 1968. p. 776.

31 FORMAN (M.L.)- Jl. of the optic. soc. of America, vol. 56, n° 7, juillet 1966.

32 KAHANER (D.K.) - IEEE trans. and. elec. vol. Au-18, n°4, décembre 1970.

33 HARTWELL (J.W.) - IBM J - Res. Develop. Sept. 1971. p.355-363.

34 COCHRAM (W.T.), COOLEY (J.W.), FAVIN (D.L.) and all - IEEE trans. aud. el. acous., vol Au-15, n° 2, June 1967.

35 CONNES (J.) - Aspen Internat. Conference - 1970.

36 DELOUIS (H.) " " " "

37 CONNES (J.) et (P.), MAILLARD (J.P.) : Atlas des spectres dans le proche infra-rouge de Vénus, Mars, Jupiter, Saturne - Edition du CNRS - 1969.

38 GUELACHVILI (G.) Thèse, Fac. des Sciences d'Orsay - Mars 1973.

39 CONNES (J.), DELOUIS (H.), GAUTIER (F.) - Brochure interne "Le Manuel du Parfait petit Fouriériste" (ci-jointe) - CIRCE - ORSAY 1972.

40 MARKEL (J.D.) - IEEE trans. aud. elec. - vol Au-19, n° 4, décembre 1971.

SOUS PROGRAMME ''FFT32T-V3''
****

```
      SUBROUTINE FFT32T(TR,TI,N,S)                                    **V3**
C
C     CE PROGR. CALCULE LA D.F.T. D'UNE FONCTION DE N ECHANTILLONS COMPLEXES.
C          N DOIT ETRE EGAL A UNE PUISSANCE DE 2
C          TR ET TI SONT DES TABLEAUX DE DIMENSION  N.
C          S EST UN TABLEAU DE DIMENSION N/2.
C  EN ENTREE-----LES ECHANT. REELS SONT RANGES DANS TR,ET LES IMAG. DANS TI.
C                LES MEMOIRES NON UTILISEES DOIVENT ETRE MISES A ZERO.
C  EN SORTIE ---- LE SPECTRE COMPLEXE DE N ECHANT. SE TROUVE DANS TR (PARTIE
C    REELLE) ET DANS TI (PARTIE IMAG.). SI H EST LE PAS D'ECHANT. EN ENTREE
C     LE PAS D'ECHANT. EN SORTIE EST 1/N*H.
C   LA PLUS PETITE VALEUR POSSIBLE POUR N EST 32 . A TITRE INDICATIF CE PROGR.
C   PREND 38 1/100 SEC. SUR UN 360/75 IBM LORSQUE N EST EGAL A 2048.
C
      DIMENSION TR(1),TI(1),S(1)
      NS16=N/16
      NS8=NS16+NS16
      NS4=NS8+NS8
      NS2=NS4+NS4
      NS2P1=NS2+1
      NS4M1=NS4-1
      NS4M2=NS4-2
      NS4P1=NS4+1
      NS4P2=NS4+2
      NS8M1=NS8-1
      NS8P1=NS8+1
      N161=NS16+1
      NS816=NS16+NS8
      N8161=NS816+1
C
C   REMPLISSAGE DE LA DEUXIEME MOITIE DU TABLEAU S AVEC LES INDICES CORRESPON
C  DANT A L'I.B. DES N/4 PREMIERS NOMBRES PREMIERS.
C  (SEULES LES NS16 PREMIERES VAL. SONT STOCKEES DANS S DE LA VAL. S(NS4+1) A
C     S(NS4+NS16)).
      S(NS4P1)=1.1
      S(NS4P2)=NS8+1.1
      IA=NS4+3
      IADD=NS16
      IC=2
   26 LOB2=NS4+IC
      DO 27 LOB=NS4P1,LOB2,2
      S(IA)=S(LOB)+IADD
      S(IA+1)=S(LOB+1)+IADD
   27 IA=IA+2
      IADD=IADD/2
      IC=IC+IC
      IF(IC.LT.NS16)GO TO 26
C
C        PREMIRE PARTIE,TRI INITIAL DES DONNEES.
C        ---------------
C
C     RANGEMENT DE TR
C  STOCKAGE DE 1/4 DU TABLEAU TR DANS LA PREMIERE MOITIE DU TABLEAU S
C   TASSEMENT
      IFI=0
      ICAR=0
   30 ICAR=ICAR+1
      IF(ICAR-2)600,601,602
C
  600 JL=N-3
      IS=NS4
      IL=JL
   44 S(IS)=TR(JL)
      IS=IS-1
      TR(IL)=TR(JL-1)
      TR(IL-1)=TR(JL-2)
      TR(IL-2)=TR(JL-3)
      IL=IL-3
      JL=JL-4
      IF(IS.NE.1)GO TO 44
      S(1)=TR(1)
      GO TO 40
C
  601 JL=N-1
      IS=NS4
      IL=JL
  644 S(IS)=TR(JL)
      IS=IS-1
      TR(IL)=TR(JL-1)
      TR(IL-1)=TR(JL-2)
      IL=IL-2
      JL=JL-3
      IF(IS.NE.1)GO TO 644
      S(1)=TR(NS4P2)
      TR(NS2P1)=TR(NS4P1)
      GO TO 40
C
  602 IF(ICAR.EQ.4)GO TO 540
      JL=N-1
      IS=NS4
      IL=JL
  744 S(IS)=TR(JL)
      IS=IS-1
      TR(IL)=TR(JL-1)
      JL=JL-2
      IL=IL-1
      IF(IS.NE.1)GO TO 744
      S(1)=TR(NS2P1)
      GO TO 40
C
  540 J=NS2+NS4P1
      DO 31 I=1,NS4,8
      S(I)=TR(J)
      S(I+1)=TR(J+1)
      S(I+2)=TR(J+2)
      S(I+3)=TR(J+3)
      S(I+4)=TR(J+4)
      S(I+5)=TR(J+5)
      S(I+6)=TR(J+6)
      S(I+7)=TR(J+7)
   31 J=J+8
C  RECLASSEMENT ET STOCKAGE DES NS4 PREMIERES VAL. DE S.
   40 DO 51 IAA=1,NS16,2
      IT=S(IAA+NS4)+IFI
      TR(IT)=S(IAA)
      TR(IT+1)=S(IAA+NS8)
      TR(IT+2)=S(IAA+NS16)
      TR(IT+3)=S(IAA+NS816)
      IT=S(IAA+NS4P1)+IFI
      TR(IT)=S(IAA+1)
      TR(IT+1)=S(IAA+NS8P1)
      TR(IT+2)=S(IAA+N161)
   51 TR(IT+3)=S(IAA+N8161)
C
      IFI=IFI+NS4
      IF(ICAR.NE.4)GO TO 30
C
C     RANGEMENT DE TI
C
      IFI=0
      ICAR=0
  230 ICAR=ICAR+1
      IF(ICAR-2)800,801,802
C
  800 JL=N-3
      IS=NS4
      IL=JL
  944 S(IS)=TI(JL)
      IS=IS-1
      TI(IL)=TI(JL-1)
      TI(IL-1)=TI(JL-2)
      TI(IL-2)=TI(JL-3)
      IL=IL-3
```

```
      JL=JL-4
      IF(IS.NE.1)GO TO 944
      S(1)=TI(1)
      GO TO 240
C
  801 JL=N-1
      IS=NS4
      IL=JL
  844 S(IS)=TI(JL)
      IS=IS-1
      TI(IL)=TI(JL-1)
      TI(IL-1)=TI(JL-2)
      IL=IL-2
      JL=JL-3
      IF(IS.NE.1)GO TO 844
      S(1)=TI(NS4P2)
      TI(NS2P1)=TI(NS4P1)
      GO TO 240
C
  802 IF(ICAR.EQ.4)GO TO 640
      JL=N-1
      IS=NS4
      IL=JL
  544 S(IS)=TI(JL)
      IS=IS-1
      TI(IL)=TI(JL-1)
      JL=JL-2
      IL=IL-1
      IF(IS.NE.1)GO TO 544
      S(1)=TI(NS2P1)
      GO TO 240
C
  640 J=NS2+NS4P1
      DO231 I=1,NS4,8
      S(I)=TI(J)
      S(I+1)=TI(J+1)
      S(I+2)=TI(J+2)
      S(I+3)=TI(J+3)
      S(I+4)=TI(J+4)
      S(I+5)=TI(J+5)
      S(I+6)=TI(J+6)
      S(I+7)=TI(J+7)
  231 J=J+8
  240 DO251 IAA=1,NS16,2
      IT=S(IAA+NS4)+IFI
      TI(IT)=S(IAA)
      TI(IT+1)=S(IAA+NS8)
      TI(IT+2)=S(IAA+NS16)
      TI(IT+3)=S(IAA+NS816)
      IT=S(IAA+NS4P1)+IFI
      TI(IT)=S(IAA+1)
      TI(IT+1)=S(IAA+NS8P1)
      TI(IT+2)=S(IAA+N161)
  251 TI(IT+3)=S(IAA+N8161)
C
      IFI=IFI+NS4
      IF(ICAR.NE.4)GO TO 230
C
C        DEUXIEME PARTIE,LE CALCUL DES INTERCOMBINAISONS.
C        ------------------
C
C   CONSTITUTION DU TABLEAU DES COSINUS
      J=NS4
      FNS2=NS2
      PIC=3.1415926535?/FNS2
      DO 188 I=1,3
      FII=I
      FIC=PIC*FII
      S(I)=COS(FIC)
  188 S(NS4-I)=SIN(FIC)
      C1=S(1)
      C2=S(2)
      C3=S(3)
      S1=S(NS4M1)
      S2=S(NS4M2)
      S3=S(NS4-3)
      NS8M4=NS8-4
      NSI=NS4
      DO 20 I=4,NS8M4,4
      FII=I
      FIC=PIC*FII
      SI=COS(FIC)
      SS=SIN(FIC)
      S(I)=SI
      NSI=NSI-4
      S(NSI)=SS
      S(I+1)=SI*C1-SS*S1
      S(I+2)=SI*C2-SS*S2
      S(I+3)=SI*C3-SS*S3
      S(NSI-1)=SS*C1+SI*S1
      S(NSI-2)=SS*C2+SI*S2
   20 S(NSI-3)=SS*C3+SI*S3
      S(NS8)=0.707107
      S(NS4)=0.0
C
C   IL Y A PNM1+1 INTERCOMBINAISONS
      I=1
  122 TRP=TR(I+1)
      TIP=TI(I+1)
      TR(I+1)=TR(I)-TRP
      TI(I+1)=TI(I)-TIP
      TR(I)=TR(I)+TRP
      TI(I)=TI(I)+TIP
      I=I+2
      TRP=TR(I+1)
      TIP=TI(I+1)
      TR(I+1)=TR(I)-TRP
      TI(I+1)=TI(I)-TIP
      TR(I)=TR(I)+TRP
      TI(I)=TI(I)+TIP
      I=I+2
      IF(I.LT.N)GO TO 122
      I=1
  123 TRP=TR(I+2)
      TIP=TI(I+2)
      TR(I+2)=TR(I)-TRP
      TI(I+2)=TI(I)-TIP
      TR(I)=TR(I)+TRP
      TI(I)=TI(I)+TIP
      I=I+1
      TRP=TR(I+2)
      TIP=TI(I+2)
      TR(I+2)=TR(I)+TIP
      TI(I+2)=TI(I)-TRP
      TR(I)=TR(I)-TIP
      TI(I)=TI(I)+TRP
      I=I+3
      IF(I.LT.N)GO TO 123
C
      L=1
  332 TRP=TR(L+4)
      TIP=TI(L+4)
      TR(L+4)=TR(L)-TRP
      TI(L+4)=TI(L)-TIP
      TR(L)=TR(L)+TRP
      TI(L)=TI(L)+TIP
      L=L+1
      TRC=TR(L+4)*0.707107
      TIC=TI(L+4)*0.707107
      TR(L+4)=TR(L)-TRC+TIC
      TI(L+4)=TI(L)-TIC-TRC
      TR(L)=TR(L)+TRC-TIC
      TI(L)=TI(L)+TIC+TRC
      L=L+1
      TRP=TI(L+4)
      TIP=TR(L+4)
      TR(L+4)=TR(L)+TRP
      TI(L+4)=TI(L)-TIP
      TR(L)=TR(L)-TRP
      TI(L)=TI(L)+TIP
```

```
      L=L+1
      TRC=TR(L+4)*0.707107
      TIC=TI(L+4)*0.707107
      TR(L+4)=TR(L)+TRC+TIC
      TI(L+4)=TI(L)-TRC+TIC
      TR(L)=TR(L)-TRC-TIC
      TI(L)=TI(L)+TRC-TIC
      L=L+5
      IF(L.LT.N)GO TO 332
C
      IDW=NS16
      K=4
   23 L=1
   22 IW=IDW
      II=L+K
      J=II+K
      JJ=J+K
      TRP=TR(J)
      TIP=TI(J)
      TR(J)=TR(L)-TRP
      TI(J)=TI(L)-TIP
      TR(L)=TR(L)+TRP
      TI(L)=TI(L)+TIP
      TRP=TI(JJ)
      TIP=TR(JJ)
      TR(JJ)=TR(II)+TRP
      TI(JJ)=TI(II)-TIP
      TR(II)=TR(II)-TRP
      TI(II)=TI(II)+TIP
   24 L=L+1
      J=J+1
      A=S(IW)
      B=S(NS4-IW)
      TRP=TR(J)*A-TI(J)*B
      TIP=TI(J)*A+TR(J)*B
      TR(J)=TR(L)-TRP
      TI(J)=TI(L)-TIP
      TR(L)=TR(L)+TRP
      TI(L)=TI(L)+TIP
      II=II+1
      JJ=JJ+1
      TRP=TR(JJ)*B+TI(JJ)*A
      TIP=TR(JJ)*A-TI(JJ)*B
      TR(JJ)=TR(II)+TRP
      TI(JJ)=TI(II)-TIP
      TR(II)=TR(II)-TRP
      TI(II)=TI(II)+TIP
      IW=IW+IDW
      IF(IW.LT.NS4)GO TO 24
      L=JJ+1
      IF(L.LT.N)GO TO 22
      IDW=IDW/2
      K=K+K
      IF(K.LT.NS4P1)GO TO 23
      RETURN
      END                                                          **V2**
```

SOUS PROGRAMME ''FFT64K-V3''

```
      SUBROUTINE FFT64K(TR,TI,N,IIR,S)                             **V3**
C  CE SSP. CALCULE LA DFT COMPLEXE B D'UNE FONCTION REELLE X ECHANT. AVEC UN
C     MAXIMUM DE ND=2*N VALEURS. (N ETANT UNE PUISSANCE DE 2)
C   DIMENSIONS DES TABLEAUX A RESERVER DANS LE PROGR. PRINCIPAL.
C              TR(N),TI(N),S(N/2)    AVEC N >16
C   EN ENTREE,LES ECHANTILLONS DE X SONT RENTRES SUCCESIVEMENT DANS TR PUIS
C     DANS TI,LES MEMOIRES NON UTILISEES ETANT MISES A ZERO.
C     DANS LE SEUL CAS OU X EST PAIRE OU IMPAIRE ON PEUT N UTILISER QUE
C     LA MOITIE DES POINTS EN FAISANT PARTIR L ECHANTILLONNAGE DU CENTRE
C     DE SYMETRIE.IL FAUT ALORS REMPLIR TI DE ZEROS
C     A L INTERIEUR DE L INTERVALLE SPECTRAL        ON TROUVE LE
C     SPECTRE ET UNE IMAGE.L ENSEMBLE EST ECHANTILLONNE EN N*2 POINTS.
C   A LA SORTIE :
C     SI IIR< OU =1,ON TROUVE DANS TR ET TI LES N PREMIERES VALEURS.
C     REELLES ET IMAGINAIRES DE LA FONCTION B
C    -SI IIR>1,CE SONT LES N DERNIERES VALEURS QU'ON Y TROUVE
C     A TITRE INDICATIF CE PROGR. PREND 24 1/100 SEC. SUR UN 360/75 IBM POUR
C     ND=2048.
C
      DIMENSION TR(1),TI(1),S(1)
C
      NS16=N/16
      NS8=NS16+NS16
      NS4=NS8+NS8
      NS2=NS4+NS4
      NS2P1=NS2+1
      NS4M1=NS4-1
      NS4M2=NS4-2
      NS4P1=NS4+1
      NS4P2=NS4+2
      NS8M1=NS8-1
      NS8P1=NS8+1
      N161=NS16+1
      NS816=NS16+NS8
      N8161=NS816+1
C
C        PREMIERE PARTIE,LE RECLASSEMENT DES DONNEES.
C        --------------------
      I2=-2
      DO 171 I=1,NS2,2
      I2=I2+4
      S(I)=TR(I2)
      TR(I)=TR(I2-1)
      S(I+1)=TR(I2+2)
  171 TR(I+1)=TR(I2+1)
      J=1
      DO 173 I=NS2P1,N,4
      TR(I)=TI(J)
      TR(I+1)=TI(J+2)
      TR(I+2)=TI(J+4)
      TR(I+3)=TI(J+6)
  173 J=J+8
      I1=N-1
      J=N-2
  174 TI(I1)=TI(J)
      TI(I1-1)=TI(J-2)
      I1=I1-2
      J=J-4
      IF(J.GT.2)GO TO 174
      TI(NS2P1)=TI(2)
      DO 175 I=1,NS2,4
      TI(I)=S(I)
      TI(I+1)=S(I+1)
      TI(I+2)=S(I+2)
  175 TI(I+3)=S(I+3)
C
C        DEUXIEME PARTIE:LE CALCUL DE LA DFTN. PAR ''FFT32T''
C        --------------------
      S(NS4P1)=1.1
      S(NS4P2)=NS8+1.1
      IA=NS4+3
      IADD=NS16
      IC=2
   26 LOB2=NS4+IC
      DO 27 LOB=NS4P1,LOB2,2
      S(IA)=S(LOB)+IADD
      S(IA+1)=S(LOB+1)+IADD
   27 IA=IA+2
      IADD=IADD/2
      IC=IC+IC
      IF(IC.LT.NS16)GO TO 26
C
      IFI=0
      ICAR=0
   30 ICAR=ICAR+1
      IF(ICAR-2)600,601,602
  600 JL=N-3
```

```
      IS=NS4
      IL=JL
   44 S(IS)=TR(JL)
      IS=IS-1
      TR(IL)=TR(JL-1)
      TR(IL-1)=TR(JL-2)
      TR(IL-2)=TR(JL-3)
      IL=IL-3
      JL=JL-4
      IF(IS.NE.1)GO TO 44
      S(1)=TR(1)
      GO TO 40
  601 JL=N-1
      IS=NS4
      IL=JL
  644 S(IS)=TR(JL)
      IS=IS-1
      TR(IL)=TR(JL-1)
      TR(IL-1)=TR(JL-2)
      IL=IL-2
      JL=JL-3
      IF(IS.NE.1)GO TO 644
      S(1)=TR(NS4P2)
      TR(NS2P1)=TR(NS4P1)
      GO TO 40
  602 IF(ICAR.EQ.4)GO TO 540
      JL=N-1
      IS=NS4
      IL=JL
  744 S(IS)=TR(JL)
      IS=IS-1
      TR(IL)=TR(JL-1)
      JL=JL-2
      IL=IL-1
      IF(IS.NE.1)GO TO 744
      S(1)=TR(NS2P1)
      GO TO 40
  540 J=NS2+NS4P1
      DO 31 I=1,NS4,8
      S(I)=TR(J)
      S(I+1)=TR(J+1)
      S(I+2)=TR(J+2)
      S(I+3)=TR(J+3)
      S(I+4)=TR(J+4)
      S(I+5)=TR(J+5)
      S(I+6)=TR(J+6)
      S(I+7)=TR(J+7)
   31 J=J+8
C
   40 DO 51 IAA=1,NS16,2
      IT=S(IAA+NS4)+IFI
      TR(IT)=S(IAA)
      TR(IT+1)=S(IAA+NS8)
      TR(IT+2)=S(IAA+NS16)
      TR(IT+3)=S(IAA+NS816)
      IT=S(IAA+NS4P1)+IFI
      TR(IT)=S(IAA+1)
      TR(IT+1)=S(IAA+NS8P1)
      TR(IT+2)=S(IAA+N161)
   51 TR(IT+3)=S(IAA+N8161)
      IFI=IFI+NS4
      IF(ICAR.NE.4)GO TO 30
C
      IFI=0
      ICAR=0
  230 ICAR=ICAR+1
      IF(ICAR-2)800,801,802
  800 JL=N-3
      IS=NS4
      IL=JL
  944 S(IS)=TI(JL)
      IS=IS-1
      TI(IL)=TI(JL-1)
      TI(IL-1)=TI(JL-2)
      TI(IL-2)=TI(JL-3)
      IL=IL-3
      JL=JL-4
      IF(IS.NE.1)GO TO 944
      S(1)=TI(1)
      GO TO 240
  801 JL=N-1
      IS=NS4
      IL=JL
  844 S(IS)=TI(JL)
      IS=IS-1
      TI(IL)=TI(JL-1)
      TI(IL-1)=TI(JL-2)
      IL=IL-2
      JL=JL-3
      IF(IS.NE.1)GO TO 844
      S(1)=TI(NS4P2)
      TI(NS2P1)=TI(NS4P1)
      GO TO 240
  802 IF(ICAR.EQ.4)GO TO 640
      JL=N-1
      IS=NS4
      IL=JL
  544 S(IS)=TI(JL)
      IS=IS-1
      TI(IL)=TI(JL-1)
      JL=JL-2
      IL=IL-1
      IF(IS.NE.1)GO TO 544
      S(1)=TI(NS2P1)
      GO TO 240
  640 J=NS2+NS4P1
      DO231 I=1,NS4,8
      S(I)=TI(J)
      S(I+1)=TI(J+1)
      S(I+2)=TI(J+2)
      S(I+3)=TI(J+3)
      S(I+4)=TI(J+4)
      S(I+5)=TI(J+5)
      S(I+6)=TI(J+6)
      S(I+7)=TI(J+7)
  231 J=J+8
  240 DO251 IAA=1,NS16,2
      IT=S(IAA+NS4)+IFI
      TI(IT)=S(IAA)
      TI(IT+1)=S(IAA+NS8)
      TI(IT+2)=S(IAA+NS16)
      TI(IT+3)=S(IAA+NS816)
      IT=S(IAA+NS4P1)+IFI
      TI(IT)=S(IAA+1)
      TI(IT+1)=S(IAA+NS8P1)
      TI(IT+2)=S(IAA+N161)
  251 TI(IT+3)=S(IAA+N8161)
      IFI=IFI+NS4
      IF(ICAR.NE.4)GO TO 230
C
      J=NS4
      FNS2=NS2
      PIC=3.14159265359/FNS2
      DO 188 I=1,3
      FII=I
      FIC=PIC*FII
      S(I)=COS(FIC)
  188 S(NS4-I)=SIN(FIC)
      C1=S(1)
      C2=S(2)
      C3=S(3)
      S1=S(NS4M1)
      S2=S(NS4M2)
      S3=S(NS4-3)
      NS8M4=NS8-4
      NS1=NS4
      DO 20 I=4,NS8M4,4
      FII=I
      FIC=PIC*FII
      S1=COS(FIC)
```

```
      SS=SIN(FIC)
      S(I)=SI
      NSI=NSI-4
      S(NSI)=SS
      S(I+1)=SI*C1-SS*S1
      S(I+2)=SI*C2-SS*S2
      S(I+3)=SI*C3-SS*S3
      S(NSI-1)=SS*C1+SI*S1
      S(NSI-2)=SS*C2+SI*S2
   20 S(NSI-3)=SS*C3+SI*S3
      S(NS8)=0.707107
      S(NS4)=0.0
C
      I=1
  122 TRP=TR(I+1)
      TIP=TI(I+1)
      TR(I+1)=TR(I)-TRP
      TI(I+1)=TI(I)-TIP
      TR(I)=TR(I)+TRP
      TI(I)=TI(I)+TIP
      I=I+2
      TRP=TR(I+1)
      TIP=TI(I+1)
      TR(I+1)=TR(I)-TRP
      TI(I+1)=TI(I)-TIP
      TR(I)=TR(I)+TRP
      TI(I)=TI(I)+TIP
      I=I+2
      IF(I.LT.N)GO TO 122
      I=1
  123 TRP=TR(I+2)
      TIP=TI(I+2)
      TR(I+2)=TR(I)-TRP
      TI(I+2)=TI(I)-TIP
      TR(I)=TR(I)+TRP
      TI(I)=TI(I)+TIP
      I=I+1
      TRP=TR(I+2)
      TIP=TI(I+2)
      TR(I+2)=TR(I)+TIP
      TI(I+2)=TI(I)-TRP
      TR(I)=TR(I)-TIP
      TI(I)=TI(I)+TRP
      I=I+3
      IF(I.LT.N)GO TO 123
      L=1
  332 TRP=TR(L+4)
      TIP=TI(L+4)
      TR(L+4)=TR(L)-TRP
      TI(L+4)=TI(L)-TIP
      TR(L)=TR(L)+TRP
      TI(L)=TI(L)+TIP
      L=L+1
      TRC=TR(L+4)*0.707107
      TIC=TI(L+4)*0.707107
      TR(L+4)=TR(L)-TRC+TIC
      TI(L+4)=TI(L)-TIC-TRC
      TR(L)=TR(L)+TRC-TIC
      TI(L)=TI(L)+TIC+TRC
      L=L+1
      TRP=TI(L+4)
      TIP=TR(L+4)
      TR(L+4)=TR(L)+TRP
      TI(L+4)=TI(L)-TIP
      TR(L)=TR(L)-TRP
      TI(L)=TI(L)+TIP
      L=L+1
      TRC=TR(L+4)*0.707107
      TIC=TI(L+4)*0.707107
      TR(L+4)=TR(L)+TRC+TIC
      TI(L+4)=TI(L)-TRC+TIC
      TR(L)=TR(L)-TRC-TIC
      TI(L)=TI(L)+TRC-TIC
      L=L+5
      IF(L.LT.N)GO TO 332
      IDW=NS16
      K=4
   23 L=1
   22 IW=IDW
      II=L+K
      J=II+K
      JJ=J+K
      TRP=TR(J)
      TIP=TI(J)
      TR(J)=TR(L)-TRP
      TI(J)=TI(L)-TIP
      TR(L)=TR(L)+TRP
      TI(L)=TI(L)+TIP
      TRP=TI(JJ)
      TIP=TR(JJ)
      TR(JJ)=TR(II)+TRP
      TI(JJ)=TI(II)-TIP
      TR(II)=TR(II)-TRP
      TI(II)=TI(II)+TIP
   24 L=L+1
      J=J+1
      A=S(IW)
      B=S(NS4-IW)
      TRP=TR(J)*A-TI(J)*B
      TIP=TI(J)*A+TR(J)*B
      TR(J)=TR(L)-TRP
      TI(J)=TI(L)-TIP
      TR(L)=TR(L)+TRP
      TI(L)=TI(L)+TIP
      II=II+1
      JJ=JJ+1
      TRP=TR(JJ)*B+TI(JJ)*A
      TIP=TR(JJ)*A-TI(JJ)*B
      TR(JJ)=TR(II)+TRP
      TI(JJ)=TI(II)-TIP
      TR(II)=TR(II)-TRP
      TI(II)=TI(II)+TIP
      IW=IW+IDW
      IF(IW.LT.NS4)GO TO 24
      L=JJ+1
      IF(L.LT.N)GO TO 22
      IDW=IDW/2
      K=K+K
      IF(K.LT.NS4P1)GO TO 23
C
C        TROISIEME PARTIE: LE CALCUL DE LA FONCTION HERMITIQUE.
C        ------------------
C
      PIC=PIC/2.
      Z=1.
      IF(IIR.GT.1)Z=-1.
      L1=1
      L2=NS4M1
      A=Z
      B=0.
      C1=COS(PIC)
      S1=SIN(PIC)
      ICO=1
      NP=N+2
      TR(1)=(TR(1)+2*TI(1))*2.
      TI(1)=0.
      TR(NS2P1)=TR(NS2P1)*2.
      TI(NS2P1)=TI(NS2P1)*Z*2.
      NN=NS2+2
      LM=NS2
      NM=N
      DO 1000 LN=2,NS4
      R1=TR(LN)+TR(NM)
      R2=TI(LN)+TI(NM)
      Y1=TI(LN)-TI(NM)
      Y2=TR(NM)-TR(LN)
      IF(ICO.EQ.0)GO TO 288
      AA=A
      BB=B
      A=A*C1-BB*S1
```

```
      R=BB*C1+AA*S1
      ICO=0
      GO TO 388
  288 A=S(L1)*Z
      B=S(L2)*Z
      L1=L1+1
      L2=L2-1
      ICO=1
  388 RC=R2*A
      YS=Y2*B
      YC=Y2*A
      RS=R2*B
      TR(LN)=R1+RC-YS
      TI(LN)=Y1+YC+RS
      TR(NM)=R1-RC+YS
      TI(NM)=YC+RS-Y1
      R1=TR(LM)+TR(NN)
      R2=TI(LM)+TI(NN)
      Y1=TI(LM)-TI(NN)
      Y2=TR(NN)-TR(LM)
      RC=R2*B
      YS=Y2*A
      YC=Y2*B
      RS=R2*A
      TR(LM)=R1+RC-YS
      TI(LM)=Y1+YC+RS
      TR(NN)=R1-RC+YS
      TI(NN)=YC+RS-Y1
      LM=LM-1
      NN=NN+1
 1000 NM=NM-1
      R1=TR(LM)+TR(NM)
      Y1=TI(LM)-TI(NM)
      RC=(TI(LM)+TI(NM))*0.707107*Z
      YS=(TR(NM)-TR(LM))*0.707107*Z
      TR(LM)=R1+RC-YS
      TI(LM)=Y1+YS+RC
      TR(NM)=R1-RC+YS
      TI(NM)=YS+RC-Y1
      RETURN
      END
```

SOUS PROGRAMME ''FFT128-V3''

```
      SUBROUTINE FFT128(TR,TI,N,IR,S)                          **V4**
C
C
C ***    CE SOUS PROGRAMME CALCULE LA D.F.T. B D'UNE FONCTION I REELLE IMPAIRE
C *** ECHANTILLONNEE A PARTIR DU POINT DE SYMETRIE AVEC M VALEURS. SOIT N LA
C *** PUISSANCE DE 2 TELLE QUE M SOIT INF. OU EGAL A 2*N. ON COMPLETE ALORS LA
C *** FONCTION I AVEC (2*N-M) ZEROS AVANT L'APPEL AU SOUS PROGRAMME FFT128. DANS
C *** LE PROG. PRINC.,LES TABLEAUX TR ET TI DOIVENT ETRE DIMENSIONNES A N.
C *** ON DOIT RENTRER DANS TR LES ECHANTILLONS I(1),I(2),....,I(N) ET DANS TI
C *** LES ECHANTILLONS I(N+1),I(N+2),.....,I(2*N) CE QUI PEUT ETRE FACILITE
C *** PAR LA RESERVATION DANS LE PROG. PRINC. D'UN TABLEAU DE DIMENSION 2*N EN
C *** EQUIVALENCE SUR TR ET TI.
C ***    EN SORTIE,LES 2*N ECHANTILLONS DU SPECTRE B SONT RANGES DE LA MEME FACON
C *** ILS CORRESPONDENT AU PREMIER INTERVALLE SPECTRAL LIBRE DE LA PERIODE SI IR
C *** =1 ET AU DEUXIEME SI IR SUP. A 1.
C *** SI H EST LE PAS DANS I,LE PAS DANS B SERA 1/(4*N*H).
C *** S EST UN TABLEAU DE MANOEUVRE DE DIMENSION N/2 UTILISE PAR FFT128 .IL DOIT
C *** FAIRE PARTIE DES TABLEAUX RESEVES PAR LE PROG. PRINC..
C *** REMARQUE. N DOIT OBLIGATOIREMENT ETRE EGAL A UNE PUISSANCE DE 2 SUP.A 16
C
      DIMENSION TR(1),TI(1),S(1)
      RN=N
      NS16=N/16
      NS8=NS16+NS16
      NS4=NS8+NS8
      NS2=NS4+NS4
      NS2P1=NS2+1
      NS4M1=NS4-1
      NS4M2=NS4-2
      NS4P1=NS4+1
      NS4P2=NS4+2
      NS8M1=NS8-1
      NS8P1=NS8+1
      N161=NS16+1
      NS816=NS16+NS8
      N8161=NS816+1
      NS2M1=NS2-1
      NS2P2=NS2+2
      PISN=3.14159265359/RN
C
C *** CONSTITUTION DE LA FCT. HERMITIQUE. N=ND/4=DIMENS. DES TBLX TR ET TI.
C     ------------------------------------
      IN1=1
      TR0=TR(1)
      TR(1)=TR(2)
      TRNP1=-TI(N)
      DO 1145 IN=1,NS2M1
      IN1=IN1+2
      S(IN)=TR(IN1)
 1145 TR(IN+1)=(TR(IN1+1)-TR(IN1-1))/2.
      S(NS2)=TI(1)
      TR(NS2P1)=(TI(2)-TR(N))/2.
      IN1=1
      DO 1146 IN=NS2P2,N
      IN1=IN1+2
      TI(IN-NS2P1)=TI(IN1)
 1146 TR(IN)=(TI(IN1+1)-TI(IN1-1))/2.
C *** ARRANGEMENT DE TI
      J=NS2P1
      TI(NS2P2)=TI(1)
      DO 1147 I=2,NS2M1,2
      J=J+2
      TI(J)=TI(I)
      TI(I)=S(I-1)
      TI(J+1)=TI(I+1)
 1147 TI(I+1)=S(I)
      TI(NS2)=S(NS2M1)
      TI(NS2P1)=S(NS2)
      TI(1)=TR0
C
C     SUBROUTINE FFT65K(TR,TI,N,TRNP1,S)
C
C *** CE PROG. CALCULE UNE FCT. REELLE B,DFT D'UNE FCT.HERMITIQUE T ECHANTIL.
C *** AVEC N*2    VALEURS COMPLEXES. LES N  PREMIERES VAL. REELLES DE T
C *** SONT RENTREES DANS TRET LES N PREMIERES VAL. IMAG. DANS TI.
C *** TR ET TI DOIVENT DONC ETRE DIMENSIONNES A N QUI EST UNE PUISSANCE
C *** DE 2. S EST  UN TABLEAU DIMENSIONNE A N/2.
C ***  TRNP1=PARTIE REELLE DE T(N+1)
C ********    EN SORTIE
C *** LA FONCTION B EST REELLE. SES N PREMIERS ECHANTILLONS SE TROUVENT DANS
C *** TR ET SES N DERNIERS DANS TI.
C
C***  ETAPE D'INTERCOMBINAISONS
C
      C1=COS(PISN)
      S1=SIN(PISN)
      A=1.
      B=0.
      ICO=0
      NN=NS2P2
      LM=NS2
      NM=N
      DO 288 LN=2,NS4
      IF(ICO.EQ.7)GO TO 1288
      ICO=ICO+1
      AA=A
      A=A*C1-B*S1
      B=B*C1+AA*S1
      GO TO 2288
 1288 RP=LN-1
      FIC=PISN*RP
      A=COS(FIC)
```

```
      B=SIN(FIC)
      ICO=0
 2288 A1=TR(LN)+TR(NM)
      A4=TI(LN)-TI(NM)
      A5=TI(LN)+TI(NM)
      A6=TR(LN)-TR(NM)
      A2=-B*A6-A*A5
      A3=A*A6-B*A5
      TR(LN)=A1+A2
      TR(NM)=A1-A2
      TI(LN)=A4+A3
      TI(NM)=A3-A4
      A1=TR(LM)+TR(NN)
      A4=TI(LM)-TI(NN)
      A5=TI(LM)+TI(NN)
      A6=TR(LM)-TR(NN)
      A2=-A*A6-B*A5
      A3=B*A6-A*A5
      TR(LM)=A1+A2
      TR(NN)=A1-A2
      TI(LM)=A4+A3
      TI(NN)=A3-A4
      LM=LM-1
      NN=NN+1
  288 NM=NM-1
      A1=TR(LM)+TR(NM)
      A4=TI(LM)-TI(NM)
      A5=-0.707107*(TI(LM)+TI(NM))
      A6=(TR(LM)-TR(NM))*0.707107
      A2=A5-A6
      A3=A5+A6
      TR(LM)=A1+A2
      TR(NM)=A1-A2
      TI(LM)=A4+A3
      TI(NM)=A3-A4
      A=TR(1)
      TR(1)=A+TRNP1
      TI(1)=A-TRNP1
      TI(NS2P1)=-TI(NS2P1)*2.
      TR(NS2P1)=TR(NS2P1)*2.
C
C *** PARTIE DFT PAR FFT32T    **V2**
C *** CALCUL DE LA TF COMPLEXE D'UNE FCT.COMPLEXE QCQ.
      S(NS4P1)=1.1
      S(NS4P2)=NS8+1.1
      IA=NS4+3
      IADD=NS16
      IC=2
   26 LOB2=NS4+IC
      DO 27 LOB=NS4P1,LOB2,2
      S(IA)=S(LOB)+IADD
      S(IA+1)=S(LOB+1)+IADD
   27 IA=IA+2
      IADD=IADD/2
      IC=IC+IC
      IF(IC.LT.NS16)GO TO 26
C
      IFI=0
      ICAR=0
   30 ICAR=ICAR+1
      IF(ICAR-2)600,601,602
  600 JL=N-3
      IS=NS4
      IL=JL
   44 S(IS)=TR(JL)
      IS=IS-1
      TR(IL)=TR(JL-1)
      TR(IL-1)=TR(JL-2)
      TR(IL-2)=TR(JL-3)
      IL=IL-3
      JL=JL-4
      IF(IS.NE.1)GO TO 44
      S(1)=TR(1)
      GO TO 40
  601 JL=N-1
      IS=NS4
      IL=JL
  644 S(IS)=TR(JL)
      IS=IS-1
      TR(IL)=TR(JL-1)
      TR(IL-1)=TR(JL-2)
      IL=IL-2
      JL=JL-3
      IF(IS.NE.1)GO TO 644
      S(1)=TR(NS4P2)
      TR(NS2P1)=TR(NS4P1)
      GO TO 40
  602 IF(ICAR.EQ.4)GO TO 540
      JL=N-1
      IS=NS4
      IL=JL
  744 S(IS)=TR(JL)
      IS=IS-1
      TR(IL)=TR(JL-1)
      JL=JL-2
      IL=IL-1
      IF(IS.NE.1)GO TO 744
      S(1)=TR(NS2P1)
      GO TO 40
  540 J=NS2+NS4P1
      DO 31 I=1,NS4,8
      S(I)=TR(J)
      S(I+1)=TR(J+1)
      S(I+2)=TR(J+2)
      S(I+3)=TR(J+3)
      S(I+4)=TR(J+4)
      S(I+5)=TR(J+5)
      S(I+6)=TR(J+6)
      S(I+7)=TR(J+7)
   31 J=J+8
   40 DO 51 IAA=1,NS16,2
      IT=S(IAA+NS4)+IFI
      TR(IT)=S(IAA)
      TR(IT+1)=S(IAA+NS8)
      TR(IT+2)=S(IAA+NS16)
      TR(IT+3)=S(IAA+NS816)
      IT=S(IAA+NS4P1)+IFI
      TR(IT)=S(IAA+1)
      TR(IT+1)=S(IAA+NS8P1)
      TR(IT+2)=S(IAA+N161)
   51 TR(IT+3)=S(IAA+N8161)
      IFI=IFI+NS4
      IF(ICAR.NE.4)GO TO 30
C
      IFI=0
      ICAR=0
  230 ICAR=ICAR+1
      IF(ICAR-2)800,801,802
  800 JL=N-3
      IS=NS4
      IL=JL
  944 S(IS)=TI(JL)
      IS=IS-1
      TI(IL)=TI(JL-1)
      TI(IL-1)=TI(JL-2)
      TI(IL-2)=TI(JL-3)
      IL=IL-3
      JL=JL-4
      IF(IS.NE.1)GO TO 944
      S(1)=TI(1)
      GO TO 240
  801 JL=N-1
      IS=NS4
      IL=JL
  844 S(IS)=TI(JL)
      IS=IS-1
      TI(IL)=TI(JL-1)
      TI(IL-1)=TI(JL-2)
      IL=IL-2
      JL=JL-3
```

```
      IF(IS.NE.1)GO TO 844
      S(1)=TI(NS4P2)
      TI(NS2P1)=TI(NS4P1)
      GO TO 240
  802 IF(ICAR.EQ.4)GO TO 640
      JL=N-1
      IS=NS4
      IL=JL
  544 S(IS)=TI(JL)
      IS=IS-1
      TI(IL)=TI(JL-1)
      JL=JL-2
      IL=IL-1
      IF(IS.NE.1)GO TO 544
      S(1)=TI(NS2P1)
      GO TO 240
  640 J=NS2+NS4P1
      DO231 I=1,NS4,8
      S(I)=TI(J)
      S(I+1)=TI(J+1)
      S(I+2)=TI(J+2)
      S(I+3)=TI(J+3)
      S(I+4)=TI(J+4)
      S(I+5)=TI(J+5)
      S(I+6)=TI(J+6)
      S(I+7)=TI(J+7)
  231 J=J+8
  240 DO251 IAA=1,NS16,2
      IT=S(IAA+NS4)+IFI
      TI(IT)=S(IAA)
      TI(IT+1)=S(IAA+NS8)
      TI(IT+2)=S(IAA+NS16)
      TI(IT+3)=S(IAA+NS816)
      IT=S(IAA+NS4P1)+IFI
      TI(IT)=S(IAA+1)
      TI(IT+1)=S(IAA+NS8P1)
      TI(IT+2)=S(IAA+N161)
  251 TI(IT+3)=S(IAA+N8161)
      IFI=IFI+NS4
      IF(ICAR.NE.4)GO TO 230
C
      J=NS4
      FNS2=NS2
      PIC=3.14159265359/FNS2
      DO 188 I=1,3
      FII=I
      FIC=PIC*FII
      S(I)=COS(FIC)
  188 S(NS4-I)=SIN(FIC)
      C1=S(1)
      C2=S(2)
      C3=S(3)
      S1=S(NS4M1)
      S2=S(NS4M2)
      S3=S(NS4-3)
      NS8M4=NS8-4
      NSI=NS4
      DO 20 I=4,NS8M4,4
      FII=I
      FIC=PIC*FII
      SI=COS(FIC)
      SS=SIN(FIC)
      S(I)=SI
      NSI=NSI-4
      S(NSI)=SS
      S(I+1)=SI*C1-SS*S1
      S(I+2)=SI*C2-SS*S2
      S(I+3)=SI*C3-SS*S3
      S(NSI-1)=SS*C1+SI*S1
      S(NSI-2)=SS*C2+SI*S2
   20 S(NSI-3)=SS*C3+SI*S3
      S(NS8)=0.707107
      S(NS4)=0.0
C
      I=1
  122 TRP=TR(I+1)
      TIP=TI(I+1)
      TR(I+1)=TR(I)-TRP
      TI(I+1)=TI(I)-TIP
      TR(I)=TR(I)+TRP
      TI(I)=TI(I)+TIP
      I=I+2
      TRP=TR(I+1)
      TIP=TI(I+1)
      TR(I+1)=TR(I)-TRP
      TI(I+1)=TI(I)-TIP
      TR(I)=TR(I)+TRP
      TI(I)=TI(I)+TIP
      I=I+2
      IF(I.LT.N)GO TO 122
      I=1
  123 TRP=TR(I+2)
      TIP=TI(I+2)
      TR(I+2)=TR(I)-TRP
      TI(I+2)=TI(I)-TIP
      TR(I)=TR(I)+TRP
      TI(I)=TI(I)+TIP
      I=I+1
      TRP=TR(I+2)
      TIP=TI(I+2)
      TR(I+2)=TR(I)+TIP
      TI(I+2)=TI(I)-TRP
      TR(I)=TR(I)-TIP
      TI(I)=TI(I)+TRP
      I=I+3
      IF(I.LT.N)GO TO 123
      L=1
  332 TRP=TR(L+4)
      TIP=TI(L+4)
      TR(L+4)=TR(L)-TRP
      TI(L+4)=TI(L)-TIP
      TR(L)=TR(L)+TRP
      TI(L)=TI(L)+TIP
      L=L+1
      TRC=TR(L+4)*0.707107
      TIC=TI(L+4)*0.707107
      TR(L+4)=TR(L)-TRC+TIC
      TI(L+4)=TI(L)-TIC-TRC
      TR(L)=TR(L)+TRC-TIC
      TI(L)=TI(L)+TIC+TRC
      L=L+1
      TRP=TI(L+4)
      TIP=TR(L+4)
      TR(L+4)=TR(L)+TRP
      TI(L+4)=TI(L)-TIP
      TR(L)=TR(L)-TRP
      TI(L)=TI(L)+TIP
      L=L+1
      TRC=TR(L+4)*0.707107
      TIC=TI(L+4)*0.707107
      TR(L+4)=TR(L)+TRC+TIC
      TI(L+4)=TI(L)-TRC+TIC
      TR(L)=TR(L)-TRC-TIC
      TI(L)=TI(L)+TRC-TIC
      L=L+5
      IF(L.LT.N)GO TO 332
      IDW=NS16
      K=4
   23 L=1
   22 IW=IDW
      II=L+K
      J=II+K
      JJ=J+K
      TRP=TR(J)
      TIP=TI(J)
      TR(J)=TR(L)-TRP
      TI(J)=TI(L)-TIP
      TR(L)=TR(L)+TRP
      TI(L)=TI(L)+TIP
      TRP=TI(JJ)
```

```
      TIP=TR(JJ)
      TR(JJ)=TR(II)+TRP
      TI(JJ)=TI(II)-TIP
      TR(II)=TR(II)-TRP
      TI(II)=TI(II)+TIP
   24 L=L+1
      J=J+1
      A=S(IW)
      B=S(NS4-IW)
      TRP=TR(J)*A-TI(J)*B
      TIP=TI(J)*A+TR(J)*B
      TR(J)=TR(L)-TRP
      TI(J)=TI(L)-TIP
      TR(L)=TR(L)+TRP
      TI(L)=TI(L)+TIP
      II=II+1
      JJ=JJ+1
      TRP=TR(JJ)*B+TI(JJ)*A
      TIP=TR(JJ)*A-TI(JJ)*B
      TR(JJ)=TR(II)+TRP
      TI(JJ)=TI(II)-TIP
      TR(II)=TR(II)-TRP
      TI(II)=TI(II)+TIP
      IW=IW+IDW
      IF(IW.LT.NS4)GO TO 24
      L=JJ+1
      IF(L.LT.N)GO TO 22
      IDW=IDW/2
      K=K+K
      IF(K.LT.NS4P1)GO TO 23
C
C *** FIN DE LA PARTIE FFT32T
C
C *** DERNIERE ETAPE D'INTERCOMBINAISON  POUR CALCULER UNE DES DEUX MOITIEES
C *** DU SPECTRE
C     CETTE ETAPE A ETE MISE AVANT LE TRIAGE QUI SUIT POUR EVITER D'AVOIR A RE-
C     CALCULER LES SINUS.
      U=+1.
      IF(IR.GT.1)U=-1.
      PISN=PISN/2.
      CI=COS(PISN)
      SI=SIN(PISN)
      A=CI*U
      B=SI*U
      C2=TI(1)-TI(N)
      C3=(TI(1)+TI(N))/B
      TI(1)=C3-C2
      TI(N)=C3+C2
C
      TR(1)=0.
      TR(NS2P1)=U*TR(NS2P1)*2.
      IS=1
      I=N
      J=1
 1230 J=J+1
      AA=A
      A=A*CI-B*SI
      B=B*CI+AA*SX
      C2=TR(J)-TR(I)
      C3=(TR(J)+TR(I))/B
      TR(J)=C3-C2
      TR(I)=C3+C2
      AA=A
      A=A*CI-B*SI
      B=B*CI+AA*SX
      I=I-1
      C2=TI(J)-TI(I)
      C3=(TI(J)+TI(I))/B
      TI(J)=C3-C2
      TI(I)=C3+C2
      IF(J.EQ.NS2)GO TO 1200
      J=J+1
      A=S(IS)*U
      B=S(NS4-IS)*U
      IS=IS+1
      C2=TR(J)-TR(I)
      C3=(TR(J)+TR(I))/B
      TR(J)=C3-C2
      TR(I)=C3+C2
      AA=A
      BB=B
      A=AA*CI-BB*SX
      B=BB*CI+AA*SX
      I=I-1
      C2=TI(J)-TI(I)
      C3=(TI(J)+TI(I))/B
      TI(J)=C3-C2
      TI(I)=C3+C2
      GO TO 1230
 1200 CONTINUE
C
C
C *** TRIAGE FINAL DE LA FCT. REELLE AINSI  OBTENUE COMME DFT. DE LA FCT.
C *** HERMITIQUE
      DO 171 I=1,NS2,4
      S(I)=TI(I)
      S(I+1)=TI(I+1)
      S(I+2)=TI(I+2)
  171 S(I+3)=TI(I+3)
      I1=NS2P1
      DO 172 I2=2,N,4
      TI(I2)=TI(I1)
      TI(I2-1)=TR(I1)
      TI(I2+2)=TI(I1+1)
      TI(I2+1)=TR(I1+1)
  172 I1=I1+2
      I1=NS2
      I3=N
  174 TR(I3)=S(I1)
      TR(I3-1)=TR(I1)
      TR(I3-2)=S(I1-1)
      TR(I3-3)=TR(I1-1)
      I1=I1-2
      I3=I3-4
      IF(I3.GT.1)GO TO 174
C
      RETURN
      END
```

- PROGRAMME ''TFID4'' -

```
//TFID4 JOB .........
//ETAP1 EXEC FTHCLG
//FORT.SYSIN DD *
C
C        *****  PROGRAMME TFID4          AVEC SIMULATION.
C
C      CE PROGR. CALCULE LA D.F.T. D'UNE FONCTION I REELLE IMPAIRE ECHANTILLON-
C    NEE A PARTIR DU POINT DE SYMETRIE AVEC UN NOMBRE M D'ECHANTILLONS,LORSQUE
C    CE NOMBRE EST TROP GRAND POUR QUE,SUR UN ORDINATEUR DONNE,LE CALCUL PUISSE
C    ETRE FAIT EN MEMOIRE CENTRALE PAR UNE METHODE FFT. CLASSIQUE.
C       SOIT N LA PUISSANCE DE 2 SUP. OU EGALE A M. ON UTILISE LA METHODE DITE
C    DE ''DECIMATION IN TIME'' POUR SE RAMENER AU CALCUL EN MEMOIRE CENTRALE DE
C    LA DFT. DE INBL SOUS-TABLEAUX ISSUS DES M VALEURS DE I.  LES RESULTATS
C    INTERMEDIAIRES SONT STOCKES SUR DISQUE MAGNETIQUE EN ACCES DIRECT.
C    LA D.F.T. EN M.C. EST CALCULEE PAR LE SSP.'FFT32T'-VERSION V31-
C       NDIMTR LA DIM. DES TBLX TR ET TI EST A FIXER EN FONCTION DE LA PLACE
C    DISPONIBLE EN MEM. CENTR.   LA VAL. DE INBL EST ALORS EGALE A N/(NDIMTR*2).
C -CHOIX DE JPL
C +++++++++++++
C      EN ENTREE      LES M VALEURS DE L'INTERF. I SONT LUES SUR LA BANDE MAGNE-
C    TIQUE FT32 PAR BLOCS DE JPL VALEURS.LE PARAMETRE JPL EST FIXE PAR L'UTILI-
C    SATEUR ET LES BLOCS SONT TRANSFERES DANS LE TABLEAU D QUI DOIT ETRE DIMEN-
C    SIONNE A (JPL+3). CES BLOCS SUR LA BANDE SONT PRECEDES D'UN ENREGISTREMENT
C    OU ON TROUVE NU,LE NUMERO DE L'INTERF. , ET M QUI EST LE NOMBRE DE VAL.. LA
C    VAL. DE M  UTILISEE POUR LE CALCUL EST LUE SUR UNE CARTE DONNEE. ELLE PEUT
C    ETRE INFER. A CELLE SE TROUVANT SUR LA BANDE ET ELLE EST RAMENEE AUTOMATI-
C    QUEMENT AU MULTIPLE DE JPL IMMEDIATEMENT INFERIEUR.
C  ON A INTERET A PRENDRE JPL GRAND.NOUS ALLONS VOIR A PROPOS DU CHOIX DE NDIMTR
C  COMMENT CE PARAMETRE INTERVIENT DANS L'ENCOMBREMENT EN M.C. DU PROGR.
C -CHOIX DE NDIMTR
C +++++++++++++++++
C    NDIMTR DOIT ETRE UNE PUISSANCE DE 2. DANS LA PRATIQUE IL N'EST PAS
C  RENTABLE DE DEPASSER 16K MOTS.
C  L'ENCOMBREMENT TOTAL EN M.C. EST DONNE EN MOTS PAR LA FORMULE.
C      2*NDIMTR + NDIMTR/4 + SUP(JPL,NDIMTR/2) + BUFFERS + PROGR.RESULTANT
C  LE PROGRAMME RESULTANT OCCUPE 9700 MOTS,MAIS POUR GAGNER DE LA PLACE,IL POUR-
C  RAIT ETRE FACILEMENT DIVISE EN PLUSIEURS ETAPES INDEPENDANTES.
C  DANS LE CHOIX DE NDIMTR,TENIR COMPTE DU FAIT QUE N DOIT ETRE SUP. A NDIMTR*2
C -CHOIX DE LSIZE
C  ++++++++++++++++
C  LES 3 FICHIERS DISQUES (FT01,FT11,FT21) SONT DEFINIS PAR TROIS INSTRUCTIONS
C  DEFINE FILE 11(N1,LSIZE,U,IV) QUI PERMETTENT LORSQUE LE FICHIER EST CREE
C  (DISP=NEW) DE RESERVER N1 BLOCS DE LSIZE MOTS SUR LE DISQUE.
C    LSIZE DOIT ETRE LE PLUS GRAND POSSIBLE AVEC DEUX RESTRICTIONS.
C       -LSIZE DOIT ETRE UNE PUISSANCE DE 2 INFER. OU EGALE A NDIMTR/4.
C       -LSIZE DOIT PERMETTRE UNE OCCUPATION OPTIMISEE DU DISQUE
C  N1 SERA CHOISI EN SACHANT QUE N NE POURRA PAS DEPASSER LA VAL. N1*LSIZE*2.
C -CONCLUSION.. EN FONCTION DES PARAMETRES CHOISIS ON AURA TOUJOURS.
C ++++++++++    M INFER. OU = A N1*LSIZE*2
C               M SUP. OU = A JPL ET M SUP. A NDIMTR*2
C  TOUTE MODIFICATION DE NDIMTR ET/OU LSIZE ENTRAINERA UNE MODIFICATION DES
C  CARTES REPEREES PAR UN */ ENTRE LES COLONNES 73 ET 80.
C -UTILISATION EVENTUELLE DU FICHIER DISQUE FT22
C  ++++++++++++++++++++++++++++++++++++++++++++++
C      AVANT LE CALCUL DE LA D.F.T. DES INBL SOUS FICHIERS,IL EST NECESSAIRE DE
C  FAIRE UN TRI PAR ETAPES EN REPARTISSANT SUR CES INBL SOUS-FICHIERS  2*NDIMTR
C  VALEURS STOCKEES EN M.C..  A CHAQUE ETAPE DE TRI,ON NE PEUT DONC STOCKER SUR
C  DISQUE QUE NPM8 VALEURS POUR CHAQUE SOUS FICHIER AVEC
C    NPM8=(NDIMTR*2)/INBL=(4*NDIMTR**2)/N ET C'EST UNE PUISSANCE DE 2.
C  SI LSIZE >NPM8 IL FAUDRA PASSER PAR UN FICHIER DISQUE INTERMEDIAIRE FT22 DONT
C  LES BLOCS SERONT DE LONGUEUR MSIZE EGALE OU INFER. A NPM8. ON PRENDRA DONC
C  POUR MSIZE LA PLUS PETITE VAL. POSSIBLE DE NPM8. CE FICHIER FT22 EST DEFINI
C  PAR L'INSTRUCTION   DEFINE FILE(NLS,MSIZE,U,IV)  AVEC.
C  -NLS=(LSIZE/MSIZE)*INBL OU INBL A LA VAL. MAX. POSSIBLE.
C    -SI LE FICHIER FT22 N'EST PAS NECESSAIRE (NPM8 TOUJOURS SUP. OU = A LSIZE),
C  ON PEUT SUPPRIMER TOUTES LES INSTRUCTIONS REPEREES PAR UN  C  EN COLONNE 80
C  POUR DIMINUER LA TAILLE DU PROGRAMME.
C -DIMENSIONS ET EQUIVALENCES.
C ++++++++++++++++++++++++++++
C   TRI(2*NDIMTR),D(JPL+3),G(LSIZE),IST(INBL MAX.),W(NDIMTR/4),S(NDIMTR/2)
C   S EN EQUIV. SUR D, G EN EQUIV. SUR W,  TR ET TI(NDIMTR) EN EQUIV. SUR TRI.
C   TXR,TXI,BXR,BXI, DE DIMENSION NDIMTR/2 EN EQUIV. SUR TRI.
C   TR1,TR2,TI1,TI2,BR1,BR2,BI1,BI2, DE DIMENSION NDIMTR/4 EN EQUIV.SUR TRI.
C
C      EN SORTIE.
C      **********          CE PROGRAMME CALCULE N VAL. DE LA TF. EN SINUS DE I
C   REPARTIES SUR UNE DES DEUX MOITIES DE SA PERIODE 1/H SI H EST LE PAS D'ECHAN
C   TILLONNAGE DE I.IL CALCULE LA PREMIERE MOITIE SI IR=1,ET LA DEUXIEME SI IR>1
C   LE PAS DANS LE SPECTRE CALCULE EST EGAL A 1/(2*N*H).
C   LES N/2 PREMIERES VALEURS SE TROUVENT DANS LE FICHIER 1,ET LES N/2 DERNIERES
C   DANS LE FICHIER 11. CES VALEURS DOIVENT ETRE LUES PAR L'INTERMEDIAIRE D'UN
C   TABLEAU DIMENSIONNE AVEC UN MULTIPLE DE LSIZE.
C  CES VAL. NON NORMALISEES SONT ULTERIEUREMENT TRAITEES PAR LE PROGR. ''T102''
C -CARACTERISTIQUES DU PROGR. TEL QU'IL EST.
C ++++++++++++++++++++++++++++++++++++++++++++
C  NDIMTR=128,LSIZE=32,MSIZE=4,JPL=50,N1=256,NLS=512
C  M DOIT ETRE COMPRIS ENTRE 256 ET 16384 BORNES EXCLUEES. LE FICHIER FT22 EST
C  NECESSAIRE DES QUE M SUP A 2048
C
C  REMARQUE.
C  *********     TOUTES  LES CARTES MARQUEES 'SIMU' EN COL.73 CORRESPONDENT A UN
C   JEU D'ESSAI PERMETTANT DE SIMULER LA LECTURE DE LA BANDE  FT32 ET D'INTRO-
C   DUIRE UNE SINUSOIDE APODISEE A DES FINS DE TESTS.
C      POUR LIRE EFFECTIVEMENT LES M VALEURS DE I SUR LA BANDE  ''FT32'',ENLEVER
C   TOUTES LES CARTES  MARQUEES ''SIMU'',ET REMETTRE EN FONCTION CELLES MARQEES
C   ''TAPE'' EN SUPPRIMANT LE ''C'' DE LA COLONNE 1.
C
C
      DIMENSION RESUL(150)                                                SIMU
      DIMENSION TR(128),TI(128),S(64),TRI(256),G(32),W(32),IST(64)       */
      DIMENSION TXR(64),TXI(64),BXR(64),BXI(64),BR2(32),BI2(32),D(53)    */
      DIMENSION TR1(32),TI1(32),TR2(32),TI2(32),BR1(32),BI1(32)          */
C
      EQUIVALENCE (TRI(1),TR(1)),(TRI(129),TI(1))                        */
      EQUIVALENCE (TR(1),TXR(1)),(TR(65),TXI(1))                         */
      EQUIVALENCE (TI(1),BXR(1)),(TI(65),BXI(1)),(W(1),G(1))             */
      EQUIVALENCE (TR(1),TR1(1)),(TR(33),TR2(1)),(TR(65),TI1(1))         */
      EQUIVALENCE (TI(1),BR1(1)),(TI(33),BR2(1)),(TI(65),BI1(1))         */
      EQUIVALENCE (TR(97),TI2(1)),(TI(97),BI2(1)),(S(1),D(1))            */
      DATA PI/3.14159265/,KERR/1/,JPL/50/,LSIZE/32/,NDIMTR/128/          */
      DEFINE FILE 1(256,32,U,IV),11(256,32,U,IV),21(256,32,U,IV)         */
      MSIZE=4                                                            */     C
      DEFINE FILE 22(512,4,U,IV)                                         */     C
C
      NU=999                                                              SIMU
      MM=16383                                                            SIMU
C                     SEULE CARTE DONNEE A LIRE.
      READ(5,782)IR,M
  782 FORMAT(I5,I10)
C     READ(32)NU,MM,NU,MM,NU,MM                                           TAPE
      IF(M.GT.MM)M=MM
      M=(M/JPL)*JPL
      PRINT 46,NU,MM,M,JPL,IR
   46 FORMAT(' NU=',I7,' M(FT32)=',I7,' M(CALCUL)=',I7,' JPL=',I6,' IR='
     1,I2,/)
      N=NDIMTR
  301 IF(N.GT.M)GO TO 300
      N=N+N
      GO TO 301
  300 RN=N
      MK1=M/JPL
      NDTRS4=NDIMTR/4
      INBL=N/(NDIMTR*2)
      IF(INBL.LT.2)GO TO 302                                                KERR=1
      NPM8=(NDIMTR*2)/INBL
      NPM82=NDIMTR/2
      NBL=N/NPM82
      NS2=N/2
      RNS2=NS2
```

```
      NS4=N/4
      RNS4=NS4
      NSEGM=NPM8/LSIZE
      NSEG4=NDIMTR/LSIZE
      NSEG2=NSEG4/2
      KERR=3                                                    KERR=3
      IF(NSEG2.LT.2)GO TO 302
      NSAGM=NSEG4/4
      NPM8S8=NPM82/INBL
      KERR=4                                                    KERR=4
      IF(NPM8S8.LT.1)GO TO 302
      PRINT 333,NDIMTR,NSEGM,NPM8,INBL,N,M,LSIZE,NPM8S8
  333 FORMAT(' NDIMTR    NSEGM    NPM8    INBL     N       M      LSIZE  NPM8
     1S8',/,1X,4I7,2I8,2I7,/)
C
      IST(1)=0
      IST(2)=INBL/2
      IF(INBL.EQ.2)GO TO 9485
      JN=3
      IJO=INBL/4
      JNA=1
 9483 JNA=JNA+JNA
      DO 9484 IBD=1,JNA
      IST(JN)=IST(IBD)+IJO
 9484 JN=JN+1
      IJO=IJO/2
      IF(IJO.GT.0)GO TO 9483
 9485 PRINT9583,(IST(I),I=1,INBL)
 9583 FORMAT(4X,8I10)
C
C *** CALCUL DE LA PREMIERE MOITIEE(CAD. NS2 VAL. COMPLEXES) DE LA FCT.
C *** HERMITIQUE H (PARTIE REELLE DANS TR1 ET PARTIE IMAG. DANS TI1)
C
      JR=1
      JIE=1
      NJIE=NS2/LSIZE
      IDEB=0
      JPLR=1
      JPL1=2
      NLU=0
      IJ2=0                                                     SIMU
   52 JPL2=JPL+JPLR
      IF(NLU.GE.MK1)GO TO 53
C SIMULATION DE LA LECTURE DE JPL VAL. SUR FT32.                SIMU
      QM=(M-1)*(M-1)                                            SIMU
      DO 80 J=1,JPL                                             SIMU
      IJ1=JPL1+J-1                                              SIMU
      FIJ2=IJ2                                                  SIMU
      IJ2=IJ2+1                                                 SIMU
   80 D(IJ1)=SIN(0.7854*FIJ2)*((1.-FIJ2*FIJ2/QM)**2)            SIMU
C               0.7854=PI/4                                     SIMU
C     READ(32)(D(J),J=JPL1,JPL2)                                TAPE
      NLU=NLU+1
      GO TO 54
   53 DO 55 J=JPL1,JPL2
   55 D(J)=0.
   54 IF(IDEB.EQ.0)D(1)=-D(3)
      IDEB=1
      NH=(JPL+JPLR-1)/2
      JPLR=(JPL+JPLR)-NH*2
      JPL1=JPLR+1
      I1=2
      DO 50 IA=1,NH
      TR(JR)=D(I1+1)-D(I1-1)
      TI(JR)=D(I1)+D(I1)
      I1=I1+2
      IF(JR.LT.NDIMTR)GO TO 51
      JR=1
      WRITE(1'JIE)TR
      WRITE(11'JIE)TI
      JIE=JIE+NSEG4
      IF(JIE.GT.NJIE)GO TO 60
      GO TO 50
   51 JR=JR+1
   50 CONTINUE
      DO 70 J=1,JPLR
      J1=JPL2-JPLR+J
   70 D(J)=D(J1)
      GO TO 52
   60 CONTINUE
      TINP1=0.
      TRNP1=-D(JPL2)*2.
C
C *** CALCUL DES NS2 VALEURS COMPLEXES DE LA FCT. F A LAQUELLE ON POURRA
C *** APPLIQUER LA DFT. CLASSIQUE PAR FFT32T
      M77=(NBL/2+1)*NSAGM+1
      ICLOU=0
      PISN=PI/RNS2
      JCQ=1
      A=COS(PISN)
      B=SIN(PISN)
      C=1.
      Q=0.
      NJ1=7
      NF=1
      NL=(NBL-1)*NSAGM+1
  501 READ(01'NF)TR1
      READ(11'NF)TI1
      READ(01'NL)TR2
      READ(11'NL)TI2
      IF(ICLOU.NE.0) GO TO 200
      ICLOU=1
      BR1(1)=TR1(1)+TRNP1
      BI1(1)=TR1(1)-TRNP1
      GO TO 201
  200 CC=C
      C=C*A-Q*B
      Q=Q*A+CC*B
      JCQ=JCQ+1
      A1=TR1(1)+TRNP1
      A4=TI1(1)-TINP1
      A5=TI1(1)+TINP1
      A6=TR1(1)-TRNP1
      A2=-Q*A6-C*A5
      A3=-Q*A5+C*A6
      BR2(1)=A1-A2
      BI2(1)=A3-A4
      WRITE(01'NF-NSAGM)BR1
      WRITE(11'NF-NSAGM)BI1
      WRITE(01'NL+NSAGM)BR2
      WRITE(11'NL+NSAGM)BI2
      BR1(1)=A1+A2
      BI1(1)=A4+A3
  201 NJ2=NDTRS4
      DO 500 JX=2,NDTRS4
      IF(JCQ.EQ.7)GO TO 342
      CC=C
      C=C*A-Q*B
      Q=Q*A+CC*B
      JCQ=JCQ+1
      GO TO 343
  342 RNJ1=NJ1
      JCQ=0
      NJ1=NJ1+8
      TETA=PISN*RNJ1
      C=COS(TETA)
      Q=SIN(TETA)
  343 A1=TR1(JX)+TR2(NJ2)
      A4=TI1(JX)-TI2(NJ2)
      A5=TI1(JX)+TI2(NJ2)
      A6=TR1(JX)-TR2(NJ2)
      A2=-Q*A6-C*A5
      A3=-Q*A5+C*A6
      BR1(JX)=A1+A2
      BR2(NJ2)=A1-A2
      BI1(JX)=A4+A3
      BI2(NJ2)=A3-A4
  500 NJ2=NJ2-1
      TRNP1=TR2(1)
      TINP1=TI2(1)
```

```
      NF=NF+NSAGM
      NL=NL-NSAGM
      IF(NF.LT.M??)GO TO 501
      BR2(1)=TRNP1*2.
      BI2(1)=-TINP1*2.
C
C*** TRIAGE ET DECOUPAGE DE LA FCT. F EN INBL SERIES DE NDIMTR VAL. COMPLEXES.
      NFICH=11
      NFI=21
      IF(NSEGM.LT.1)GO TO 38                                                    C
      NTGM=(NDIMTR/NPM8)*NSEGM                                                  C
      NTGM1=NTGM+1                                                              C
      DO 344 I=1,INBL                                                           C
  344 IST(I)=IST(I)*NTGM                                                        C
  484 NW=1                                                                      C
      NL=1                                                                      C
  482 I=1                                                                       C
      DO 481 IJK=1,4                                                            C
      READ(NFICH'NL)S                                                           C
      J=1                                                                       C
      NL=NL+NSEG2                                                               C
      DO 480 IJ=1,NPM8S8                                                        C
      IPI=I                                                                     C
      DO 34 IA=1,INBL                                                           C
      TRI(IPI)=S(J)                                                             C
      J=J+1                                                                     C
   34 IPI=IPI+NPM8                                                              C
  480 I=I+1                                                                     C
  481 CONTINUE                                                                  C
      I1=1                                                                      C
      I2=NPM8                                                                   C
      DO 9486 IA=1,INBL                                                         C
      WRITE(NFI'IST(IA)+NW)(TRI(I),I=I1,I2)                                     C
      I1=I1+NPM8                                                                C
 9486 I2=I2+NPM8                                                                C
      NW=NW+NSEGM                                                               C
      IF(NW.LT.NTGM1)GO TO 482                                                  C
      IF(NFICH.EQ.1)GO TO 483                                                   C
      NFICH=1                                                                   C
      NFI=11                                                                    C
      GO TO 484                                                                 C
   38 NP=LSIZE/NPM8                                                             C
      NM=NPM8/MSIZE                                                             C
      NPM=NP*NM                                                                 C
      NTGM=NDIMTR/LSIZE                                                         C
      NTGM1=NTGM+1                                                              C
      DO 35 I=1,INBL                                                            C
   35 IST(I)=IST(I)*NTGM                                                        C
   39 NW=1                                                                      C
      NL=1                                                                      C
   36 DO 32 I32=1,NP                                                            C
      I=1                                                                       C
      N31=(I32-1)*NM+1                                                          C
      DO 37 IJK=1,4                                                             C
      READ(NFICH'NL)S                                                           C
      J=1                                                                       C
      NL=NL+NSEG2                                                               C
      DO 40 IJ=1,NPM8S8                                                         C
      IPI=I                                                                     C
      DO 41 IA=1,INBL                                                           C
      TRI(IPI)=S(J)                                                             C
      J=J+1                                                                     C
   41 IPI=IPI+NPM8                                                              C
   40 I=I+1                                                                     C
   37 CONTINUE                                                                  C
      I1=1                                                                      C
      I2=NPM8                                                                   C
      DO 31 IA=1,INBL                                                           C
      WRITE(22'N31)(TRI(I),I=I1,I2)                                             C
      I1=I1+NPM8                                                                C
      I2=I2+NPM8                                                                C
   31 N31=N31+NPM                                                               C
   32 CONTINUE                                                                  C
      DO 33 IA=1,INBL                                                           C
      READ(22'(IA-1)*NPM+1)G                                                    C
   33 WRITE(NFI'IST(IA)+NW)G                                                    C
      NW=NW+1                                                                   C
      IF(NW.LT.NTGM1)GO TO 36                                                   C
      IF(NFICH.EQ.1)GO TO 483                                                   C
      NFICH=1                                                                   C
      NFI=11                                                                    C
      GO TO 39                                                                  C
C
C *** CALCUL DE LA DFT DE CHACUNE DE CES INBL SERIES PAR ''FFT32T''-V31-
  483 IWW=1
      IKP1=1
      DO 490 IA=1,INBL
      READ(11'IKP1)TR
      READ(21'IKP1)TI
      CALL FFT32T(TR,TI,NDIMTR,S,W,IWW)                                         **V31**
      IWW=0
      WRITE(11'IKP1)TR
      WRITE(21'IKP1)TI
  490 IKP1=IKP1+NSEG4
C
C *** IPUI SERIES D'INTERCOMBINAISONS PERMETTANT DE RECONSTITUER TOUT LE
C *** SPECTRE.
      NIJK=INBL
      IDIT=INBL/2
      NB=2
      PIS=PI/RNS4
C *** IL Y A IPUI ETAPES D'INTERCOMBINAISONS
  553 NB=NB+NB
      NBAG=NB*NSAGM
      NIJK=NIJK/2
      IT=0
      NFI1=1
      NFI2=NBAG+1
      FIDIT=IDIT
      PISE=PIS*FIDIT
      JCQ=8
      F=COS(PISE)
      E=SIN(PISE)
      DO 552 IJK=1,NIJK
C *** IL Y A NIJK INTERCOMBINAISONS DE 2 BLOCS 2 A 2 DANS CHAQUE ETAPE
      DO 551 IJ=1,NB
C *** IL Y A NB  BLOCS LOGIQUES DE NPM8 VALEURS PAR BLOC A INTERCOMBINER
      READ(11'NFI1)TR1
      READ(21'NFI1)TI1
      READ(11'NFI2)TR2
      READ(21'NFI2)TI2
      DO 550 I=1,NDTRS4
C *** IL Y A NDTRS4 VALEURS DANS CHAQUE BLOC LOGIQUE LU
      IF(JCQ.LT.8)GO TO 345
      JCQ=1
      FIT=IT
      TETA=PISE*FIT
      IT=IT+8
      C=COS(TETA)
      Q=SIN(TETA)
      GO TO 346
  345 CC=C
      C=C*F-Q*E
      Q=Q*F+CC*E
      JCQ=JCQ+1
  346 A=C*TR2(I)-Q*TI2(I)
      B=Q*TR2(I)+C*TI2(I)
      BR1(I)=TR1(I)+A
      BI1(I)=TI1(I)+B
      BR2(I)=TR1(I)-A
  550 BI2(I)=TI1(I)-B
      WRITE(11'NFI1)BR1
      WRITE(21'NFI1)BI1
      WRITE(11'NFI2)BR2
      WRITE(21'NFI2)BI2
      NFI1=NFI1+NSAGM
  551 NFI2=NFI2+NSAGM
      NFI1=NFI1+NBAG
      NFI2=NFI2+NBAG
  552 IT=0
```

```
      IDIT=IDIT/2
      IF(IDIT.GT.0)GO TO 553
C
C***  TRIAGE FINAL APRES FFT32T
C***  TR SE TROUVERA SUR 1   ET TI SE TROUVERA SUR 11.
      NFI1=11
      NFI2=21
      NFI3=1
      NB=NS4/NPM82
      NF=1
  682 NF2=1
      DO 680 IA=1,NB
      READ(NFI1'NF)TXR
      READ(NFI2'NF)TXI
      J=1
      DO 670 I=1,NPM82,4
      TI(J)=TXR(I)
      TI(J+1)=TXI(I)
      TI(J+2)=TXR(I+1)
      TI(J+3)=TXI(I+1)
      TI(J+4)=TXR(I+2)
      TI(J+5)=TXI(I+2)
      TI(J+6)=TXR(I+3)
      TI(J+7)=TXI(I+3)
  670 J=J+8
      WRITE(NFI3'NF2)TI
      NF2=NF2+NSEG4
  680 NF=NF+NSEG2
      IF(NFI3.EQ.11)GO TO 681
      NFI3=11
      GOTO 682
  681 CONTINUE
C
C *** DERNIERE INTERCOMBINAISON POUR OBTENIR UNE DES 2 MOITIE DE LA PERIODE 1/H.
      INBL2=INBL*2
      NFF=INBL2*NSEG2+1
      U=+1.
      IF(IA.GT.1)U=-1.
      ICLOU=0
      PISA=PI/AN
      NJ1=1
      JCQ=10
      F=COS(PISA)
      E=SIN(PISA)
      NF=1
      NL=(INBL2-1)*NSEG2+1
 1501 READ(1'NF)TXR
      READ(11'NL)TXI
      IF(ICLOU.NE.0)GO TO 1200
      ICLOU=1
      BXR(1)=0.
      GO TO 1201
 1200 CC=C
      C=C*F-Q*E
      Q=Q*F+CC*E
      JCQ=JCQ+1
      C2=TXR(1)-TINP1
      C3=(TXR(1)+TINP1)/Q
      BXI(1)=C3+C2
      WRITE(01'NF-NSEG2)BXR
      WRITE(11'NL+NSEG2)BXI
      BXR(1)=C3-C2
 1201 NJ2=NPM82
      DO 1500 JX=2,NPM82
      IF(JCQ.EQ.10)GO TO 347
      CC=C
      C=C*F-Q*E
      Q=Q*F+CC*E
      JCQ=JCQ+1
      GO TO 348
  347 RNJ1=NJ1
      NJ1=NJ1+10
      JCQ=1
      TETA=PISA*RNJ1
      C=COS(TETA)*U
      Q=SIN(TETA)*U
  348 C2=TXR(JX)-TXI(NJ2)
      C3=(TXR(JX)+TXI(NJ2))/Q
      BXR(JX)=C3-C2
      BXI(NJ2)=C3+C2
 1500 NJ2=NJ2-1
      TINP1=TXI(1)
      NF=NF+NSEG2
      NL=NL-NSEG2
      IF(NF.EQ.NFF)GO TO 1600
      GO TO 1501
 1600 BXI(1)=U*TINP1*2.
      WRITE(01'NF-NSEG2)BXR
      WRITE(11'NL+NSEG2)BXI
C                                                                        SIMU
C   SORTIE DES RESULTATS DE LA SIMULATION                                SIMU
  332 FORMAT(1X,10F12.4)                                                 SIMU
      NN1=(NS4-40)/LSIZE                                                 SIMU
      READ(01'NN1)RESUL                                                  SIMU
      PRINT 332,RESUL                                                    SIMU
      N600=150 .                                                         SIMU
      CALL COURB(N600,V1,V2,RESUL)                                       SIMU
C                                                                        SIMU
      GO TO 303
  302 PRINT 304,KERR
  304 FORMAT(' *** ARRET AU COURS DU STEP GO *** ERREUR DE TYPE'I3,/)
  303 PRINT 99
   99 FORMAT(' FIN DU TFID4',//)
      STOP
      END
/*.
//GO.FT01F001 DD UNIT=DISK,DISP=(NEW,DELETE),SPACE=(TRK,(20,2))
//GO.FT11F001 DD UNIT=DISK,DISP=(NEW,DELETE),SPACE=(TRK,(20,2))
//GO.FT21F001 DD UNIT=DISK,DISP=(NEW,DELETE),SPACE=(TRK,(20,2))
//GO.FT22F001 DD UNIT=DISK,DISP=(NEW,DELETE),SPACE=(TRK,(10,2))
//GO.SYSIN DD *
    1      8200
/*.
//
```

```
C
C    ********  PROGRAMME  '' MPLOT '' DE TRACES DE SPECTRES. ****************
C                           --------
      REAL*8 SIG1B,DELTAB,DELTAG,BENGTH,DELDX,FKKK
      REAL*8FRANGE,DX,DELTAD,ABC,DX10,RN,RXP1,SIG1G
      REAL*8 RIFR,TRANCH,RMR,RNSB,DSIGMA,RNBLOC,SIGOG,RKPM1
      DIMENSION B(1050),AA(109),G(5120),CB(1024)
      DIMENSION TEXT(12),P(5120),IBUF(1013)
      EQUIVALENCE (P(1),G(1)),(B(21),CB(1))
      DATA NADR/1/,IPEN/1/,KLIG/0/,IR/1/,MP/201/
      CALL IBENA(IBUF,1013,14)
      CALL PNUMA(0.,0.,NADR,0.,0.)
      READ (16)NU,NU,NU,ND1024,SIG1B,DELTAB
      NPRIM=ND1024*1024
      RNSB=NPRIM
      SIGB=SIG1B+RNSB*DELTAB/2.
      PRINT 1024,NU,ND1024,NPRIM,SIG1B,DELTAB
 1024 FORMAT(' NU=',I6,' ND1024=',I6,' NPRIM=',I8,' SIG1B=',D15.8,' DELT
     1AB=',D15.8,/)
      READ(5,197)FRANGE,IFR,M,IAP
  197 FORMAT(D20.13,I5,I10,I1)
      PRINT 56,FRANGE,IFR,SIGB,IAP,M
   56 FORMAT(' FRANGE=',D20.13,' IFR=',I5,' SIGB=',D15.8,' IAP=',I1,' M=
     1',I9)
      RIFR=IFR
      DELTAD=FRANGE*RIFR
      TRANCH=1./(2.*DELTAD)
      MR=SIGB*DELTAD
      RMR=MR
      DSIGB=SIGB-(RMR/DELTAD)
      IF(DSIGB.GT.TRANCH)IR=3
      DELTAB=1./(2.*DELTAD*RNSB)
      DELTAG=DELTAB/5.
      BENGTH=5120.*DELTAG
      SIG1B=RMR/DELTAD
      IF(IR.NE.1)SIG1B=SIG1B+TRANCH
      PRINT 5,DELTAD,MR,IR,SIG1B,DELTAB,BENGTH,DELTAG
    5 FORMAT(' DELTAD=',D15.8,' MR=',I5,' IR=',I5,' SIG1B=',D15.8,/,' DE
     1LTAB=',D15.8,' BENGTH=',D15.8,' DELTAG=',D15.8,/)
C
C ***  CALCUL DE LA FONCTION D INTERPOLATION AA.
C ***  AA=TF. D UN RECT. DE LONG. 1 ECHANT. AVEC  201 VAL.(DELTAD=0.005)
C ***  CE RECT. PEUT ETRE PONDERE SUR MP=( 201*M)/NPRIM VALEURS SI IAP=1 OU PLUS.
      DO 1 I=2,200
    1 B(I)=1.0
      B(201)=0.5
C   *** PONDERATION DE L INTERF
      IF(IAP.EQ.0)GO TO 12
      MP=(201*M)/NPRIM
      Q=(MP-1)*(MP-1)
      IF(IAP-2)1935,1936,1937
 1935 R=0.                                                          APO1
      S=0.
      T=1.
      W=0.
      V=2.
      GO TO 5000
 1936 R=0.06267                                                     APO2
      S=0.33213
      T=0.21839
      W=0.38680
      V=2.
      GO TO 5000
 1937 IF(IAP-3)1939,1939,1940
 1939 R=0.23978                                                     APO3
      S=0.45808
      T=0.22499
      W=0.077208
      V=2.
      GO TO 5000
 1940 R=0.                                                          APO4
      S=0.
      T=1.
      W=0.
      V=5./2.
 5000 DO 9 I=2,MP
      FI=I-1
      F=1.-(FI*FI)/Q
      PK=R+S*F+T*F**V+W*F**3
    9 B(I)=B(I)*PK
C ***  CALCUL TF CLASSIQUE  AA=TF.(B)
   12 AMAX=0.5
      DO 4 IK=2,MP
    4 AMAX=AMAX+B(IK)
      SIA=+0.3141592650-02
      DO 7 IJ=56,105
      FIJ=IJ-55
      SIG=SIA*FIJ
      AA(IJ)=0.5
      DO 3 IK=2,MP
      FIK=IK-1
    3 AA(IJ)=AA(IJ)+B(IK)*COS(SIG*FIK)
C ***  NORMALISATION ET PONDERATION DE LA FONCTION AA
    7 AA(IJ)=(AA(IJ)/AMAX)*(1.-(FIJ*FIJ)/2500.)**2
      AA(55)=1.
      DO 8 I=106,109
    8 AA(I)=0.
C ***  DEFINITION PAR SYMETRIE DE LA FCT. D'INTERP. AA COMPLETE
      DO 181 I=1,54
  181 AA(I)=AA(110-I)
      PRINT 183,MP
  183 FORMAT(' FCT. D INTERPOL. AA AVEC MP=',I5,/)
C
C ***CARACTERISTIQUES DU TRACE QUI DOIT ETRE FAIT DU BLOC NB1 OU SE TROUVE
C ***  SIG1 A CELUI OU SE TROUVE SIG2. LE TRACE EST DECOUPE EN NPLOT PLOTS
C ***  DE NBLOC BLOCS
      READ(5,2)SIG1,SIG2,NBLOC,IPLO,IFICH, DX,GMAX,TEXT, HP,NLIG,IDILAT
    2 FORMAT(2D15.8,3I5,/,D15.8,E10.3,12A4,/,E10.3,2I5)
      PRINT 185,SIG1,SIG2,NBLOC,IPLO,IFICH
  185 FORMAT(' SIG1=',D15.8,' SIG2=',D15.8,' NBLOC=',I5,' IPLO=',I5,' IF
     1ICH=',I5,/)
      IF(SIG1.GT.0.1E+10.OR.SIG1.LT.0.) GO TO 508
      IF(HP.GT.70.)HP=70.
      HT=HP/NLIG
      PRINT 212,HP,NLIG,IDILAT,HT,DX,GMAX,TEXT
  212 FORMAT(' HP=',E10.3,' NLIG=',I5,' IDILAT=',I5,' HT=',E10.3,/,' DX=
     1',D15.8,' GMAX=',E10.3,12A4)
      DILAT=IDILAT
      IF(IPLO.EQ.8)IPEN=0
      DELDX=DELTAG/DX
      HM=HT-HT/30.
      HS4=HT/13.5
      H4S5=HT/1.35
      HS5=HT/6.75
      HS52=HS5+HS4
      HS52S2=HS4+HS5/2.
      HS5M3=HS4-0.2
      HS5M5=HS4-0.4
      HS5M10=HS4-0.8
      H4S50=H4S5*DILAT/GMAX
      H4S5R=H4S5/GMAX
      HTEXT=HT/1.2
      HM0I=H4S5/2.+HS4
      H1=HS4+H4S5
      NB1=(SIG1-SIG1B)/BENGTH+1.
      NB2=(SIG2-SIG1B)/BENGTH+1.
      IF(NB1.LT.1)NB1=1
      IF(NB2.GT.ND1024)NB2=ND1024
      PRINT 14,NB1,NB2
   14 FORMAT(' NB1=',I6,' NB2=',I6,/)
      IF(NB1.EQ.1) GO TO 187
      NB1M1=NB1-1
      DO 188 I=1,NB1M1
  188 READ(16)CB
      DO 189 I=1025,1044
      J=I-1024
  189 B(J)=B(I)
      GO TO 190
```

```
  191 B(I)=0.0
  190 RNB1M1=NB1-1
C *** CALCUL ET TRACE DE NB BLOCS DE 5120 VALEURS BLOQUEES PAR GROUPES
C     DE NBLOC BLOCS
C *** LE NOMBRE DE BLOCS A TRACER SOIT NB EST ARRONDI AU MULTIPLE DE NBLOC
C *** IMMEDIATEMMENT SUP.SI CETTE VALEUR EST TROP GRANDE ,ON PREND NB=NB-NBLOC
C *** SI NB<NBLOC,LE PROGRAMME IMPOSE NBLOC=NB
C
      NB=NB2-NB1+1
      IF(NBLOC.LE.NB)GO TO 206
      NBLOC=NB
      PRINT 207,NBLOC
  207 FORMAT(' NBLOC IMPOSE PAR LE PG.',I8,/)
  206 NBB=(NB/NBLOC)*NBLOC
      IF(NB.NE.NBB)NB=NBB+NBLOC
      NLU=NB1-1+NB
      IF(NLU.GT.ND1024)NB=NB-NBLOC
      KPLOT=NB/NBLOC
      RNBLOC=NBLOC
      DSIGMA=RNBLOC*BENGTH
      AXEX=DSIGMA/DX
      XPAGE=AXEX+6.
      DX10=DX*10.
      SIGOG=SIG1B+RNB1M1*BENGTH-10.*DELTAB
C
      DO 192 KP=1,KPLOT
      KLIG=KLIG+1
      RKPM1=KP-1
      SIG1G=SIGOG+RKPM1*DSIGMA
      PRINT 208,SIG1G
  208 FORMAT(' SIG1G=',D15.8)
       CALL PCARA(7.,HTEXT,0,TEXT,48,0.2,0.3,1.,0.)
       CALL TRAA(0.,0.,0)
      KKK=0
C
      DO 193 KB=1,NBLOC
      READ(16)CB
C
C     CONVOLUTION ET INTERPOLATION.
      DO 194 K=1,5120
      I=(K-1)/5+1
      J=I*5-K+1
  194 G(K)=B(I)*AA(J)+B(I+20)*AA(J+100)+P(I+1)*AA(J+5)+B(I+19)*AA(J+95)+
     1B(I+2)*AA(J+10)+B(I+18)*AA(J+90)+B(I+3)*AA(J+15)+B(I+17)*AA(J+85)+
     2B(I+4)*AA(J+20)+B(I+16)*AA(J+80)+B(I+5)*AA(J+25)+B(I+15)*AA(J+75)+
     3B(I+6)*AA(J+30)+B(I+14)*AA(J+70)+B(I+7)*AA(J+35)+B(I+13)*AA(J+65)+
     4B(I+8)*AA(J+40)+B(I+12)*AA(J+60)+B(I+9)*AA(J+45)+B(I+11)*AA(J+55)+
     5B(I+10)*AA(J+50)
      DO 196 I=1,20
      J=I+1024
  196 B(I)=B(J)
C
C    TRACE D UN BLOC DE 5120 VALEURS INTERPOLEES
      DO 35 J=1,5120
      FKKK=KKK
      X=FKKK*DELDX
      KKK=KKK+1
      Y=P(J)*H4S5R+HS4
      IF(Y.GT.HT)Y=HT
      IF(Y.LT.0.)Y=0.
   35 CALL TRAA(X,Y,IPEN)
      IF(IPLO.LT.8)GO TO 193
C
C  MEME TRACE MAIS DILATE EN AMPLITUDE PAR UN FACTEUR IDILAT.
      KKK=KKK-5120
      FKKK=KKK
      XF=X
      YF=Y
      DEBU=FKKK*DELDX
      FIN=DEBU+5120.*DELDX
      CALL TRAA(FIN,HS52,0)
      CALL TRAA(DEBU,HS52,1)
      Y=P(1)*H4S5O+HS52
      IF (Y.GT.HM)Y=HM
      IF(Y.LT.1.)Y=1.
      CALL TRAA(DEBU,Y,0)
      DO 44 J=2,5120
      FKKK=KKK
      X=FKKK*DELDX
      KKK=KKK+1
      Y=P(J)*H4S5O+HS52
      IF(Y.GT.HM)Y=HM
      IF(Y.LT.HS52S2)Y=HS52S2
   44  CALL TRAA(X,Y,1)
       CALL TRAA(XF,YF,0)
  193 CONTINUE
C
C    TRACE DE L AXE ET DES ABCISSES
      CALL TRAA(AXEX,HS4,0)
      CALL TRAA(-0.5,HS4,1)
      IF(SIG1G)38,39,40
   40 NXP1=SIG1G+1.
      RXP1=NXP1
      DIC=(RXP1-SIG1G)/DX
      GO TO 41
   39 RXP1=0.
      DIC=0.
      GO TO 41
   38 NXP1=SIG1G
      RXP1=NXP1
      DIC=(RXP1-SIG1G)/DX
   41 CALL TRAA(0.,HS4,1)
      CALL TRAA(0.,HMOI,1)
      CALL NOMBA(-1.5,HMOI,0,0.5,1,0.2,0.3,1.,0.)
      CALL TRAA(-0.5,HMOI,0)
       CALL TRAA(0.,HMOI,1)
      CALL TRAA(0.,H1,1)
       CALL NOMBA(-1.5,H1,0,1.0,1,0.2,0.3,1.,0.)
      CALL TRAA(-0.5,H1,0)
       CALL TRAA(0.,H1,1)
      CALL TRAA(DIC,HS4,0)
      NREF=(AXEX-DIC)/10.
      NQS=0
      CALL TRAA(DIC,HS5M5,1)
       CALL NOMBA(DIC,HS5M10,0,RXP1,1,0.2,0.3,1.,0.)
      DO 36 N=1,NREF
      RN=N
      ABC=RXP1+RN*DX10
      DO 37 NQ=1,9
      NQS=NQS+1
      RNQS=NQS
      X=DIC+RNQS
       CALL TRAA(X,HS4,0)
   37 CALL TRAA(X,HS5M3,1)
      NQS=NQS+1
      RNQS=NQS
      X=DIC+RNQS
      CALL TRAA(X,HS4,0)
      CALL TRAA(X,HS5M5,1)
   36  CALL NOMBA(X,HS5M10,0,ABC,2,0.2,0.3,1.,0.)
C
C     REINITIALISATION
      IF(KLIG.EQ.NLIG)GO TO 210
      CALL PNUMA(0.,HT,NADR,0.,0.)
      GO TO 192
  210 YPAGE=-HT*(KLIG-1)
      CALL PNUMA(XPAGE,YPAGE,NADR,0.,0.)
      KLIG=0
  192 CONTINUE
C
  508 NADR=-900-IFICH
      CALL PNUMA(0.,0.,NADR,0.,0.)
      FI=(KPLOT/NLIG)*XPAGE/100.
      PRINT 42,FI
   42 FORMAT(/,'  TRACE TERMINE  (',F9.2,' METRES.)')
      STOP
      END
```

```
C  ******PROGRAMME  BSS10  ********************                    CBSS10
C
C     CE PROG. CALCULE TOUT L'INTERVALLE SPECTRAL (1/2*DELTAD OU SE TROUVE
C SIGB)D'UN INTERF. IMPAIR,LE PREMIER ECHANTILLON SE TROUVANT SUR LE POINT
C DE SYMETRIE. CET INTERF. PEUT ETRE LU SOIT SUR UNE TAPE (BD16,DEN=2).
C SOIT SUR UNE BS(''FT12'',BD16,BINAIRE). LA VALEUR MAXIMUM DU NOMBRE M D'
C ECHANTILLONS EST FIXE PAR NPRIMM QUI EST UNE PUISSSANCE DE 2. NPRIM,LE
C NOMBRE DE POINTS DE SORTIE DU SPECTRE,EST UNE PUISSANCE DE 2 SUP. OU =A M,
C MAIS INFERIEURE OU EGALE A NPRIMM.
C
      REAL*8 SIG1B,DELTAB,DELTAG,BENGTH,FKKK,DELDX
      REAL*8FRANGE,DX,DELTAD,ABC,DX10,RN,RXP1,SIG1G
      REAL*8 RIFR,TRANCH,RMR,RNSB,DSIGMA,RNBLOC,SIG0G,RKPM1
      DIMENSION WORD(15),TEXT(12),AA(109),IBUF(1013),D(1050),FAG(170)
C     DIMENSION P(4096),B(4096),BR(2048),BI(2048),STRI(1024)            4K
      DIMENSION P(16384),B(16384),BR(8192),BI(8192),STRI(4096)          16K
C     DIMENSION P(65536),B(65536),BR(32768),BI(32768),STRI(16384)       64K
      DIMENSION G(5120)
C     EQUIVALENCE(P(1),BR(1)),(P(2049),BI(1)),(D(1),STRI(1))            4K
      EQUIVALENCE (P(1),BR(1)),(P(8193),BI(1)),(G(1),STRI(1))           16K
C     EQUIVALENCE(P(1),BR(1)),(P(32769),BI(1)),(STRI(1),G(1))           64K
      EQUIVALENCE (P(1),B(1))
      DATA NADR/1/,IPEN/1/,IR/1/,MP/201/,BMAX/0./,LC1/0/,GMAX/1./
C     NPRIMM=4096                                                       4K
      NPRIMM=16384                                                      16K
C     NPRIMM=65536                                                      64K
      NPRIM=512
      NS=170
      READ(5,601)(WORD(J),J=1,15),ICHEN1,IC6,IAP,IMP,ISIGN
      PRINT610,(WORD(J),J=1,15),ICHEN1,IC6,IAP,IMP,ISIGN
  601 FORMAT(15A4,6I1)
  610 FORMAT(1X,15A4,' ICHEN1=',I1,' IC6=',I1,' IAP=',I1,' IMP=',I1,' IS
     1IGN=',I1,/)
      READ(5,2)SIGB,M,IFR,NU
    2 FORMAT(D15.8,I10,2I5)
  101 NPRIM=NPRIM*2
      IF(NPRIM.GE.M)GO TO 100
      GO TO 101
  100 JPL1=M/NS
      M=JPL1*NS
      FRANGE=+0.791123932D-05                                           HAWAI
      PRINT603,IFR,SIGB,NPRIM,M,NU,FRANGE
  603 FORMAT(' IFR=',I6,' SIGB=',D15.8,' NPRIM=',I8,' M=',I8,' NU=',I6,'
     1 FRANGE DE REF.=',D20.13)
      IF(NPRIM.LE.NPRIMM)GO TO 510
      PRINT 505
  505 FORMAT('    VALEUR DE M TROP GRANDE POUR LE PROGR. CHOISI')
      GO TO 508
  510 IF(ICHEN1.EQ.0)GOTO 63
C
C
C      LECTURE DE LA TAPE.
C
      L1=1
      N170=0
      IEOF=0
      DO 908 K=1,JPL1
      N170=N170+1
      CALL HERLEC(FAG,IMP,N170,IEOF)
      IF(IEOF.NE.9999)GO TO 147
      M=(N170-1)*NS
      PRINT 148,M
  148 FORMAT(//,'  NOUVELLE VALEUR DE M=',I7,//)
      GO TO 103
  147 IF(ISIGN.GT.0)GO TO 113
      DO 135 J=1,NS
      P(L1)=FAG(J)
  135 L1=L1+1
      GO TO 908
  113 DO 112 J=1,NS
      P(L1)=-FAG(J)
  112 L1=L1+1
  908 CONTINUE
C 103 REWIND 13                                                         LECFORT.
  103 KVRET=-1                                                          LECPLASS
      CALL CPLASS(NB,KVRET)                                             LECPLAS
      GO TO 64
C
C
C         LECTURE DE LA FT12.
   63 READ(12)NU,MM,NU,MM,NU,MM
      IF(M-10000)68,68,69
   68 READ(12)(P(J),J=1,M)
      GO TO 64
   69 MK1=M/10000
      MK2=M-MK1*10000
      IB=0
      DO 65 I=1,MK1
      IA=IB+1
      IB=IB+10000
   65 READ(12)(P(K),K=IA,IB)
      I3=MK1*10000+1
      IF(MK2.EQ.0)GO TO 64
      READ(12)(P(K),K=I3,M)
   64 IF(IMP.NE.2)GO TO 21
      M1=M/100
      K1=1
      K2=10
      DO 1040 I=1,M1
      PRINT 609,(P(K),K=K1,K2)
      K1=K1+100
 1040 K2=K2+100
  609 FORMAT(10E12.3)
C
C        CALCUL DE LA TF  (M<NPRIM)
   21 RIFR=IFR
      DELTAD=FRANGE*RIFR
      TRANCH=1./(2.*DELTAD)
      MR=SIGB*DELTAD
      RMR=MR
      DSIGB=SIGB-(RMR/DELTAD)
      IF(DSIGB.GT.TRANCH)IR=3
      PRINT 2205,DELTAD,MR,IR,NPRIM
 2205 FORMAT(' DELTAD=',D15.8,' MR=',I5,' IR=',I5,' NPRIM=',I7)
      MFP1=M+1
      DO 2704 J=MFP1,NPRIM
 2704 P(J)=0.0
      NMA=NPRIM/2
      IF(NPRIM.EQ.NPRIMM)GO TO 90
      DO 2703 J=1,NMA
      JJ=NMA+J
 2703 BI(J)=P(JJ)
   90 CALL FFT128(BR,BI,NMA,IR,STRI)
      IF(NPRIM.EQ.NPRIMM)GO TO 91
      DO 1703 J=1,NMA
      JJ=NMA+J
 1703 B(JJ)=BI(J)
   91 ND1024=NPRIM/1024
      RNSB=NPRIM
      DELTAB=1./(2.*DELTAD*RNSB)
      SIG1B=RMR/DELTAD
      IF(IR.NE.1)SIG1B=SIG1B+TRANCH
C
C   NORMALISATION A 1 DES  VALEURS PRIMAIRES B DU SPECTRE.
      DO 2708 I=1,NPRIM
      IF(B(I).LE.BMAX)GO TO 2708
      BMAX=B(I)
      IMAX=I
 2708 CONTINUE
      PRINT 2710,BMAX,IMAX
 2710 FORMAT(' BMAX=',E10.3,' IMAX=',I5)
      DO 2711 L=1,NPRIM,2
      B(L)=B(L)/BMAX
 2711 B(L+1)=B(L+1)/BMAX
      PRINT 15,(B(I),I=1,90)
   15 FORMAT(/,'  PREMIERES VALEURS DU SPECTRE',/,(10F9.3))
      IX=IMAX-89
      IY=IMAX+90
```

```
      PRINT 16,(B(I),I=IX,IY)
   16 FORMAT(/,'  VALEURS DU SPECTRE AU VOISINAGE DE BMAX',/,(10F7.3))
C
      IF(IC6.EQ.0)GO TO 512
      NBA=1
      NBZ=1024
      WRITE(16)NU,NU,NU,ND1024,SIG1B,DELTAB
      DO 513I=1,ND1024
      WRITE(16)(B(K),K=NBA,NBZ)
      NBA=NBA+1024
  513 NBZ=NBZ+1024
      PRINT 514
  514 FORMAT(' VALEURS B ECRITES SUR FT16')
C
C *** CALCUL DE LA FONCTION D INTERPOLATION AA.
C *** AA=TF. D UN RECT. DE LONG. L=1 ECHANT. AVEC  201 VAL.(DELTAD=0.005)
C *** CE RECT. PEUT ETRE PONDERE SUR MP=( 201*M)/NPRIM VALEURS SI IAP=1
  512 DO 1 I=2,200
    1 D(I)=1.0
      D(201)=0.5
      IF(IAP.EQ.0)GO TO 12
      MP=(201*M)/NPRIM
      Q=(MP-1)*(MP-1)
      DO 9 I=2,MP
      FI=I-1
    9 D(I)=D(I)*(1.-(FI*FI)/Q)**2
C *** CALCUL TF CLASSIQUE  AA=TF.(D)
   12 AMAX=0.5
      DO 4 IK=2,MP
    4 AMAX=AMAX+D(IK)
      SIA=+0.3141592653D-02
      DO 7 IJ=56,105
      FIJ=IJ-55
      SIG=SIA*FIJ
      AA(IJ)=0.5
      DO 3 IK=2,MP
      FIK=IK-1
    3 AA(IJ)=AA(IJ)+D(IK)*COS(SIG*FIK)
C *** NORMALISATION ET PONDERATION DE LA FONCTION AA
    7 AA(IJ)=(AA(IJ)/AMAX)*(1.-(FIJ*FIJ)/2500.)**2
      AA(55)=1.
      DO 8 I=106,109
    8 AA(I)=0.
C *** DEFINITION PAR SYMETRIE DE LA FCT. D'INTERP. AA COMPLETE
      DO 181 I=1,54
  181 AA(I)=AA(110-I)
C   CALCUL DU COEF. DE NORMALISATION DU SPECTRE APRES INTERPOLATION
      AN=0.
      DO 1664 I=55,105,5
 1664 AN=AN+AA(I)
      AN=AN*1.3
      PRINT 183,MP,AN,(AA(I),I=55,104)
  183 FORMAT(/,' FCT. D INTERPOLATION AA AVEC MP=',I5,'  ET COEF. DE NOR
     1MALISATION AN=',F8.4,/,(10F12.6))
C
C****       INTERPOLATION  ET  TRACE DU SPECTRE
C
      CALL IBENA(IBUF,1013,14)
      CALL PNUMA(0.,0.,NADR,0.,0.)
      DELTAG=DELTAB/5.
      BENGTH=5120.*DELTAG
      PRINT 1024,NU,SIG1B,DELTAB,DELTAG,BENGTH
 1024 FORMAT(' NU=',I6,' SIG1B=',D15.8,' DELTAB=',D15.8,' DELTAG=',D15.8
     1,' BENGTH=',D15.8)
C
C ***CARACTERISTIQUES DU TRACE QUI DOIT ETRE FAIT DU BLOC NB1 OU SE TROUVE
C *** SIG1 A CELUI OU SE TROUVE SIG2. LE TRACE EST DECOUPE EN NPLOT PLOTS
C *** DE NBLOC BLOCS
      READ (5,184)SIG1,SIG2,NBLOC,IPLO,IFICH,IGMAX
  184 FORMAT(2D15.8,4I5)
      PRINT 185,SIG1,SIG2,NBLOC,IPLO,IFICH,IGMAX
  185 FORMAT('  SIG1=',D15.8,' SIG2=',D15.8,' NBLOC=',I5,'  IPLO=',I5,'
     1IFICH=',I5,' IGMAX=',I5)
      READ(5,198)DX,HT,TEXT
  198 FORMAT(D15.8,E10.3,12A4)
      IF(IGMAX.EQ.1)READ(5,6)GMAX
    6 FORMAT(E10.3)
      PRINT 55,DX,HT,TEXT,GMAX
   55 FORMAT(' DX=',D15.8,' HT=',E10.3,12A4,' GMAX=',E10.3)
      IF(IPLO.EQ.8)IPEN=0
      DELDX=DELTAG/DX
      HM=HT-HT/30.
      HS4=HT/13.5
      H4S5=HT/1.35
      H4SN=H4S5/(AN*GMAX)
      H4SM=H4SN*10.                                                    ********
      HS5=HT/6.75
      HS52=HS5+HS4
      HS5M3=HS4-0.3
      HS5M4=HS4-0.4
      HS5M5=HS4-0.5
      HS5M10=HS4-1.
      HM0I=H4S5/2.+HS4
      H1=HS4+H4S5
      NB1=(SIG1-SIG1B)/BENGTH+1.
      NB2=(SIG2-SIG1B)/BENGTH+1.
      IF(NB1.LT.1)NB1=1
      IF(NB2.GT.ND1024)NB2=ND1024
      PRINT 14,NB1,NB2
   14 FORMAT(' NB1=',I6,' NB2=',I6,/)
      IF(NB1.EQ.1) GO TO 187
      NB1M1=NB1-1
      LC1=LC1+NB1M1*1024
      DO189 J=1,20
      I=LC1-20+J
  189 D(J)=B(I)
      GO TO 190
  187 DO 191 I=1,20
  191 D(I)=0.0
  190 RNB1M1=NB1-1
C *** CALCUL ET TRACE DE NB BLOCS DE 5120 VALEURS BLOQUEES PAR GROUPES
C     DE NBLOC BLOCS
C *** LE NOMBRE DE BLOCS A TRACER SOIT NB EST ARRONDI AU MULTIPLE DE NBLOC
C *** IMMEDIATEMENT SUP.SI CETTE VALEUR EST TROP GRANDE ,ON PREND NB=NB-NBLOC
C *** SI NB<NBLOC,LE PROGRAMME IMPOSE NBLOC=NB
      NB=NB2-NB1+1
      IF(NBLOC.LE.NB)GO TO 206
      NBLOC=NB
      PRINT 207,NBLOC
  207 FORMAT(' NBLOC IMPOSE PAR LE PG.',I8)
  206 NBB=(NB/NBLOC)*NBLOC
      IF(NB.NE.NBB)NB=NBB+NBLOC
      NLU=NB1-1+NB
      IF(NLU.GT.ND1024)NB=NB-NBLOC
      KPLOT=NB/NBLOC
      RNBLOC=NBLOC
      DSIGMA=RNBLOC*BENGTH
      DX10=DX*10.
      AXEX=DSIGMA/DX
      SIG0G=SIG1B+RNB1M1*BENGTH-10.*DELTAB
C
      DO 192 KP=1,KPLOT
      RKPM1=KP-1
      SIG1G=SIG0G+RKPM1*DSIGMA
      PRINT 208,SIG1G
  208 FORMAT(' SIG1G=',D15.8)
       CALL PCARA(7.,HT,0,TEXT,48,0.2,0.4,1.,0.)
       CALL TRAA(0.,0.,0)
      KKK=0
C
      DO 193 KB=1,NBLOC
C     READ(16)(B(J),J=21,1044)
      DO 741 J=1,1024
      JC1=J+LC1
  741 D(J+20)=B(JC1)
      LC1=LC1+1024
C
C     CONVOLUTION
```

```
      DO 194 K=1,5120
      I=(K-1)/5+1
      J=I*5-K+1
  194 G(K)=D(I)*AA(J)+D(I+20)*AA(J+100)+D(I+1)*AA(J+5)+D(I+19)*AA(J+95)+
     1D(I+2)*AA(J+10)+D(I+18)*AA(J+90)+D(I+3)*AA(J+15)+D(I+17)*AA(J+85)+
     2D(I+4)*AA(J+20)+D(I+16)*AA(J+80)+D(I+5)*AA(J+25)+D(I+15)*AA(J+75)+
     3D(I+6)*AA(J+30)+D(I+14)*AA(J+70)+D(I+7)*AA(J+35)+D(I+13)*AA(J+65)+
     4D(I+8)*AA(J+40)+D(I+12)*AA(J+60)+D(I+9)*AA(J+45)+D(I+11)*AA(J+55)+
     5D(I+10)*AA(J+50)
      DO 196 I=1,20
  196 D(I)=D(I+1024)
C
C     TRACE D UN BLOC DE 5120 VALEURS INTERPOLEES
      DO 35 J=1,5120
      FKKK=KKK
      X=FKKK*DELDX
      KKK=KKK+1
      Y=G(J)*H4SN+HS4                                                    ********
      IF(Y.GT.HT)Y=HT
      IF(Y.LT.0.)Y=0.
   35 CALL TRAA(X,Y,IPEN)
      IF(IPLO.LT.8)GO TO 193
C  MEME TRACE MAIS DILATE EN AMPLITUDE PAR UN FACTEUR 10
      KKK=KKK-5120
      FKKK=KKK
      XF=X
      YF=Y
      DEBU=FKKK*DELTAG/DX
      FIN=DEBU+(5120.*DELTAG)/DX
      CALL TRAA(FIN,HS52,0)
       CALL TRAA(DEBU,HS52,1)
      IPLU=0
      DO 44 J=1,5120,2
      FKKK=KKK
      X=FKKK*DELDX
      Y=G(J)*H4SM+HS52                                                   ********
      IF(Y.LT.HM.AND.Y.GT.1.)CALL TRAA(X,Y,IPLU)
      IPLU=1
      Y=G(J+1)*H4SM+HS52
      IF(Y.LT.HM.AND.Y.GT.1.)CALL TRAA(X+DELDX,Y,IPLU)
   44 KKK=KKK+2
       CALL TRAA(XF,YF,0)
  193 CONTINUE
C
C     TRACE DE L AXE ET DES ABCISSES
      CALL TRAA(AXEX,HS4,0)
       CALL TRAA(0.,HS4,1)
      IF(SIG1G)38,39,40
   40 NXP1=SIG1G+1.
      RXP1=NXP1
      DIC=(RXP1-SIG1G)/DX
      GO TO 41
   39 RXP1=0.
      DIC=0.
      GO TO 41
   38 NXP1=SIG1G
      RXP1=NXP1
      DIC=(RXP1-SIG1G)/DX
   41 CALL TRAA(0.,HS4,1)
      CALL TRAA(0.,HM0I,1)
      CALL NOMBA(-1.5,HM0I,0,0.5,1,0.2,0.4,1.,0.)
      CALL TRAA(-0.5,HM0I,0)
       CALL TRAA(0.,HM0I,1)
      CALL TRAA(0.,H1,1)
       CALL NOMBA(-1.5,H1,0,1.0,1,0.2,0.4,1.,0.)
      CALL TRAA(-0.5,H1,0)
       CALL TRAA(0.,H1,1)
      CALL TRAA(DIC,HS4,0)
      NREF=(AXEX-DIC)/10.
      NQS=0
      CALL TRAA(DIC,HS5M4,1)
       CALL NOMBA(DIC,HS5M10,0,RXP1,1,0.2,0.4,1.,0.)
      DO 36 N=1,NREF
      RN=N
      ABC=RXP1+RN*DX*10
      DO 37 NQ=1,9
      NQS=NQS+1
      RNQS=NQS
      X=DIC+RNQS
       CALL TRAA(X,HS4,0)
   37 CALL TRAA(X,HS5M3,1)
      NQS=NQS+1
      RNQS=NQS
      X=DIC+RNQS
      CALL TRAA(X,HS4,0)
      CALL TRAA(X,HS5M5,1)
   36  CALL NOMBA(X,HS5M10,0,ABC,2,0.2,0.4,1.,0.)
C     REINITIALISATION
  192 CALL PNUMA(AXEX+6.,0.,NADR,0.,0.)
C
      NADR=-900-IFICH
      CALL PNUMA(0.,0.,NADR,0.,0.)
      FI=KPLOT*(AXEX+6.)/100.
      PRINT 42,FI
   42 FORMAT(/,' TRACE TERMINE (',F7.2,' METRES.)')
  508 STOP
      END
```

```
      SUBROUTINE SP5
C   PAR UN FILTRAGE NUMERIQUE,CETTE PARTIE DU PROGRAMME ''TEST8'' PERMET LE CAL-
C   CUL AUTOMATIQUE DE LA TRANCHE SPECTRALE DE NPRIM VAL. PRIMAIRES QUI CONTIENT
C   L'ABCISSE SIGB LUE SUR CARTE DONNEE. LA TRANCHE CALCULEE EST 'NFM' FOIS PLUS
C   ETROITE QUE L'INTERVALLE SPECTRAL LIBRE.
C   POUR AVOIR UN CALCUL DE TOUT L'INTERVALLE SPECTRAL (NFM=1),IL FAUT QUE M
C   SOIT INFER. A NPRIM*0.9+20
C LA FCT. D'INTERP. S'APPUIE DE CHAQUE COTE DE SON C. DE SYM.SUR NNINT PTS PRIM.
C     M=MK1*JPL+MK2 OU  JPL=VAL QCQ ET M= NBRE DE VOL SUR LA B5
      REAL*8 DELTAG,BENGTH,DX,SIG1G,RNSB,DSIGMA
      REAL*8 RNBLOC,SIGOG,RKPM1,DELTA,RNF,RMR,FKKK,DELDX,RIFR
      REAL*8 FRANGE,EPSILO,DELTAD,SIG1B,DELTAB
      COMMON FRANGE,SIGB(3),M,MK1,MK2,JPL,IFR,NUMERO,IMP,ISIGN,IAP
      DIMENSION TEXT(12),IBUF(1013),AA(109),G(5120),D(10505),S(505)
C     DIMENSION B(2048),STRI(512),BR(1024),BI(1024),D501(10000)          2K
      DIMENSION B(4096),STRI(1024),BR(2048),BI(2048),D501(10000)         4K
C     DIMENSION B(16384),STRI(4096),BR(8192),BI(8192),D501(10000)       16K
C     DIMENSION B(65536),STRI(16384),BR(32768),BI(32768),D501(10000)    64K
C     EQUIVALENCE (BR(1),B(1)),(BI(1),B(1025))                           2K
      EQUIVALENCE (BR(1),B(1)),(BI(1),B(2049))                          4K
C     EQUIVALENCE (BR(1),B(1)),(BI(1),P(8193))                          16K
C     EQUIVALENCE (BR(1),B(1)),(BI(1),B(32769))                         64K
      EQUIVALENCE (D(2049),G(1)),(D(502),D501(1))
C     NPRIM=2048                                                         2K
      NPRIM=4096                                                        4K
C     NPRIM=16384                                                       16K
C     NPRIM=65536                                                       64K
      EPSILO=+0.1D-15
      RIFR=IFR
      DELTAD=FRANGE*RIFR
      PRINT 13,NUMERO,IFR,NPRIM,SIGB(1),M,DELTAD
   13 FORMAT(' NUMERO=',I6,' IFR=',I5,' NPRIM=',I6,' SIGB(1)=',F9.2,' M=
     1',I7,' DELTAD=',D15.8)
      NFM=10*(M-20)/(NPRIM*9)+1
      NNINT=20*NFM
      IF(NNINT.GT.250)NNINT=250
      RNF=NFM
      PRINT 2,NFM,NNINT
    2 FORMAT(' FACTEUR DE REDUCTION=NFM=',I5,' NNINT=',I5)
C
C   CALCUL DE LA FONCTION D'INTERPOLATION S(K) QUI PERMET LE FILTRAGE NUMERIQUE.
      NNINP1=NNINT+1
      NNINP2=NNINT+2
      NNINP3=NNINT+3
      MNINP2=(NNINT*2)+2
```

```
      DELTA=DELTAD*RNF
      TRANCH=1./(2.*DELTA)
      C=TRANCH*3.14159265
      SIGBB=SIGB(1)
      MR=SIGBB*DELTA
      RMR=MR
      DSIGB=SIGBB-(RMR/DELTA)
      IR=1
      IF(DSIGB.GT.TRANCH)IR=3
      PRINT 5, TRANCH,MR,IR
    5 FORMAT(' TRANCHE CALCULEE.(LARGEUR=',F9.2,' CM-1,MR=',I5,' IR=',I5
     1,')')
      FNMR=(4*MR)+IR
      CC=C*FNMR
      KK=0
      RNP1=NNINP1
      DFKM=-RNP1*DELTAD+EPSILO
      DO 62 K=1,MNINP2
      FKK=KK
      FKM=DFKM+FKK*DELTAD
      ARC=C*FKM
      KK=KK+1
   62 S(K)=(SIN(ARC)/ARC)*COS (CC*FKM)
      IF(EPSILO.LT.+0.1E-14)S(NNINP2)=1.
C   PONDERATION DE S(K).
      RNNIP2=NNINP2
      WI=DELTAD*RNNIP2
      DO 60 I=1,NNINP1
      KI=NNINP2-I
      FKI=KI
      WIP=(FKI*DELTAD)-EPSILO
   60 S(I)=S(I)*((1.-(WIP/WI)**2)**2)
      S(NNINP2)=S(NNINP2)*((1.-(EPSILO/WI)**2)**2)
      DO 61 I=NNINP3,MNINP2
      KI=I-NNINP2
      FKI=KI
      WIP=(FKI*DELTAD)+EPSILO
   61 S(I)=S(I)*((1.-(WIP/WI)**2 )**2)
C
      READ(12)NU,MW,NU,MW,NU,MW
      J2=JPL+501
      IF(MK1.EQ.0)GO TO 1662
      READ(12)D501
      GO TO 1663
 1662 J2=MK2+501
      READ(12)(D(J),J=502,J2)
 1663 REWIND 12
      READ(12)NU,MW,NU,MW,NU,MW
      DO 7001 I1=503,1003
 7001 D(1004-I1)=-D(I1)
C
C     INTERPOLATION INTERFEROGRAMME
      KALIRE=MK1+1
      IF(MK2.EQ.0)KALIRE=MK1
      LFIRST=1
      NAV=501
      DO 1001 KA=1,KALIRE
      IF(KA.EQ.KALIRE.AND.MK2.NE.0)GO TO 1660
      READ(12)D501
      GO TO 1661
 1660 J2=MK2+501
      READ(12)(D(J),J=502,J2)
 1661 N1=J2-NAV-NNINP1
      NN1= N1
      N1=N1/NFM+1
      NHE=NN1-(N1-1)*NFM
C
C     CALCUL DE B(L) POUR L=(LFIRST,LFIRST+N1-1)
      L=LFIRST
      N20=NAV-NNINP1
      KF=(N1-1)*NFM+N20
      DO 1002 K1=N20,KF,NFM
      K=K1+1
      B(L)=D(K)*S(1)+D(K+1)*S(2)+D(K+2)*S(3)+D(K+3)*S(4)+D(K+4)*S(5)+
     1   D(K+5)*S(6)+D(K+6)*S(7)+D(K+7)*S(8)+D(K+8)*S(9)+D(K+9)*S(10)+
     2D(K+10)*S(11)+D(K+11)*S(12)+D(K+12)*S(13)+D(K+13)*S(14)+D(K+14)*
     3S(15)+D(K+15)*S(16)+D(K+16)*S(17)+D(K+17)*S(18)
      K=K+16
      DO 1003 J=19,MNINP2,2
      K=K+2
 1003 B(L)=B(L)+D(K)*S(J)+D(K+1)*S(J+1)
 1002 L=L+1
C
      LFIRST=LFIRST+N1
      NHE=501-NRP-MNINP2+NFM
      NAV=NNINP1+NHE
      D(1)=D(10001)
      DO 1004 I=2,500,2
      D(I)=D(I+10000)
 1004 D(I+1)=D(I+10001)
 1001 CONTINUE
      MF=LFIRST-1
      PRINT 6,MF,(B(K),K=1,10)
    6 FORMAT(/,' 10 PREMIERES VAL. DE L INTERF. SECOND. DE',I7,' VALEURS
     1.',/,10E12.4)
C
C     CALCUL DE LA TRANSFORMEE DE FOURIER PAR FFT128.
      MFP1=MF+1
      DO 2704 J=MFP1,NPRIM
 2704 B(J)=0.0
      NPRIS2=NPRIM/2
      CALL FFT128(BR,BI,NPRIS2,IR,STRI)
      RNSB=NPRIM
      DELTAB=1./(2.*DELTA*RNSB)
      SIG1B=RMR/DELTA
      IF(IR.NE.1)SIG1B=SIG1B+(1./(2.*DELTA))
C
C   NORMALISATION A 1 DES  VALEURS PRIMAIRES B DU SPECTRE.
      BMAX=0.0
      DO 2708 I=1,NPRIM
      IF(B(I).LE.BMAX)GO TO 2708
      BMAX=B(I)
      IMAX=I
 2708 CONTINUE
      DO 2711 L=1,NPRIM,2
      B(L)=B(L)/BMAX
 2711 B(L+1)=B(L+1)/BMAX
      PRINT 15,BMAX,IMAX,(B(I),I=1,40)
   15 FORMAT(/,' PREMIERES VAL. DU SPECTRE.(BMAX=',E12.5,' IMAX=',I6,')'
     1,/,(18F7.3))
      IX=IMAX-89
      IY=IMAX+90
      PRINT 16,(B(I),I=IX,IY)
   16 FORMAT(//,'  VALEURS DU SPECTRE AU VOISINAGE DE BMAX',/,(18F7.3))
      ND1024=NPRIM/1024
C
C  ANCIENNE PARTIE SP6 .      (INTERPOLATION ET TRACE DE LA TRANCHE)
C---------------------------
      LC1=0
C *** CALCUL DE LA FONCTION D INTERPOLATION AA.
C *** AA=TF. D UN RECT. DE LONG. L=1 ECHANT. AVEC  201 VAL.(DELTAD=0.005)
      MP=201
      DO 1 I=2,200
    1 D(I)=1.0
      D(201)=0.5
C   PONDERATION DE L'INTERF.
      IF(IAP.EQ.0)GO TO 12
      MP=(201*MF)/NPRIM
      Q=(MP-1)*(MP-1)
      DO 9 I=2,MF
      FI=I-1
    9 D(I)=D(I)*(1.-(FI*FI)/Q)**2
C *** CALCUL TF CLASSIQUE  AA=TF.(D)
   12 AMAX=0.5
      DO 4 IK=2,MP
    4 AMAX=AMAX+D(IK)
      SIA=+0.31415926E0D-02
      DO 7 IJ=56,105
```

```
      FIJ=IJ-55
      SIG=SIA*FIJ
      AA(IJ)=0.5
      DO 3 IK=2,MP
      FIK=IK-1
    3 AA(IJ)=AA(IJ)+D(IK)*COS(SIG*FIK)
C *** NORMALISATION ET PONDERATION DE LA FONCTION AA
    7 AA(IJ)=(AA(IJ)/AMAX)*(1.-(FIJ*FIJ)/2500.)**2
      AA(55)=1.
      DO 8 I=106,109
    8 AA(I)=0.
C *** DEFINITION PAR SYMETRIE DE LA FCT. D'INTERP. AA COMPLETE
      DO 181 I=1,54
  181 AA(I)=AA(110-I)
      PRINT 183,MP
  183 FORMAT(/,' FCT. D INTERP. AA  MP=',I5)
C
C****       INTERPOLATION  ET  TRACE DU SPECTRE
C ***CARACTERISTIQUES DU TRACE QUI DOIT ETRE FAIT DU BLOC NB1 OU SE TROUVE
C *** SIG1 A CELUI OU SE TROUVE SIG2. LE TRACE EST DECOUPE EN NPLOT PLOTS
C *** DE NBLOC BLOCS
C
      CALL IBENA(IBUF,1013,14)
      NADR=1
      CALL PNUMA(0.,0.,NADR,0.,0.)
      IPEN=1
      DELTAG=DELTAB/5.
      BENGTH=5120.*DELTAG
      PRINT 10,NU,SIG1B,DELTAB,BENGTH,DELTAG
   10 FORMAT(' NU=',I6,' SIG1B=',D15.8,' DELTAB=',D15.8,' BENGTH=',E12.5
     1,' DELTAG=',E12.5)
      READ (5,184)SIG1,SIG2,NBLOC,IPLO,IFICH, DX,HT,TEXT
  184 FORMAT(2D15.8,3I5,/,D15.8,E10.3,12A4)
      PRINT 185,SIG1,SIG2,NBLOC,IPLO,IFICH,DX,HT,TEXT
  185 FORMAT(' SIG1=',D15.8,' SIG2=',D15.8,' NBLOC=',I5,' IPLO=',I5,' IF
     1ICH=',I5,/,' DX=',D15.8,' HT=',E10.3,12A4)
      IF(IPLO.EQ.8)IPEN=0
      DELDX=DELTAG/DX
      HS4=HT/13.5
      H4S5=HT/1.35
      H4SN=H4S5/2.
      H4SM=H4SN*10.                                                    *******
      HS52=HT/6.75+HS4
      HS5M3=HS4-0.3
      HS5M5=HS4-0.5
      HS5M10=HS4-1.
      NB1=(SIG1-SIG1B)/BENGTH+1.
      NB2=(SIG2-SIG1B)/BENGTH+1.
      IF(NB1.LT.1)NB1=1
      IF(NB2.GT.ND1024)NB2=ND1024
      PRINT 14,NB1,NB2
   14 FORMAT(' NB1=',I5,' NB2=',I5)
      IF(NB1.EQ.1) GO TO 187
      LC1=LC1+(NB1-1)*1024
      DO189 J=1,20
      I=LC1-20+J
  189 D(J)=B(I)
      GO TO 190
  187 DO 191 I=1,20
  191 D(I)=0.0
  190 RNB1M1=NB1-1
C *** CALCUL ET TRACE DE NB BLOCS DE 5120 VALEURS BLOQUEES PAR GROUPES
C     DE NBLOC BLOCS
C *** LE NOMBRE DE BLOCS A TRACER SOIT NB EST ARRONDI AU MULTIPLE DE NBLOC
C *** IMMEDIATEMENT SUP.SI CETTE VALEUR EST TROP GRANDE ,ON PREND NB=NB-NBLOC
C *** SI NB<NBLOC,LE PROGRAMME IMPOSE NBLOC=NB
      NB=NB2-NB1+1
      IF(NBLOC.LE.NB)GO TO 206
      NBLOC=NB
      PRINT 207,NBLOC
  207 FORMAT(' NBLOC IMPOSE PAR LE PG.',I5)
  206 NBB=(NB/NBLOC)*NBLOC
      IF(NB.NE.NBB)NB=NBB+NBLOC
      IF(NB1+NB-1.GT.ND1024)NB=NB-NBLOC
      KPLOT=NB/NBLOC
      RNBLOC=NBLOC
      DSIGMA=RNBLOC*BENGTH
      AXEX=DSIGMA/DX
      DX10=DX*10.
      SIGOG=SIG1B+RNB1M1*BENGTH-10.*DELTAB
C
      DO 192 KP=1,KPLOT
      RKPM1=KP-1
      SIG1G=SIGOG+RKPM1*DSIGMA
      PRINT 208,SIG1G
  208 FORMAT(1X,'SIG1G='D15.8)
       CALL PCARA(7.,HT,0,TEXT,48,0.2,0.4,1.,0.)
       CALL TRAA(0.,0.,0)
      KKK=0
C
      DO 193 KB=1,NBLOC
C     READ(16)(B(J),J=21,1044)
      DO 741 J=1,1024
      JC1=J+LC1
  741 D(J+20)=B(JC1)
      LC1=LC1+1024
C
C     CONVOLUTION
      DO 194 K=1,5120
      I=(K-1)/5+1
      J=I*5-K+1
  194 G(K)=D(I)*AA(J)+D(I+20)*AA(J+100)+D(I+1)*AA(J+5)+D(I+19)*AA(J+95)+
     1D(I+2)*AA(J+10)+D(I+18)*AA(J+90)+D(I+3)*AA(J+15)+D(I+17)*AA(J+85)+
     2D(I+4)*AA(J+20)+D(I+16)*AA(J+80)+D(I+5)*AA(J+25)+D(I+15)*AA(J+75)+
     3D(I+6)*AA(J+30)+D(I+14)*AA(J+70)+D(I+7)*AA(J+35)+D(I+13)*AA(J+65)+
     4D(I+8)*AA(J+40)+D(I+12)*AA(J+60)+D(I+9)*AA(J+45)+D(I+11)*AA(J+55)+
     5D(I+10)*AA(J+50)
      DO 196 I=1,20
  196 D(I)=D(I+1024)
C
C    TRACE D UN BLOC DE 5120 VALEURS INTERPOLEES
      DO 35 J=1,5120
      FKKK=KKK
      X=FKKK*DELDX
      KKK=KKK+1
      Y=G(J)*H4SN+HS4 -                                                *******
      IF(Y.GT.HT)Y=HT
      IF(Y.LT.0.)Y=0.
   35 CALL TRAA(X,Y,IPEN)
      IF(IPLO.LT.8)GO TO 193
C  MEME TRACE MAIS DILATE EN AMPLITUDE PAR UN FACTEUR 10
      KKK=KKK-5120
      FKKK=KKK
      XF=X
      YF=Y
      DEBU=FKKK*DELDX
      FIN=DEBU+BENGTH/DX
      CALL TRAA(FIN,HS52,0)
       CALL TRAA(DEBU,HS52,1)
      IPLU=0
      DO 44 J=1,5120
      FKKK=KKK
      X=FKKK*DELDX
      KKK=KKK+1
      Y=G(J)*H4SM+HS52                                                 *******
      IF(Y.GT.HT)Y=HT
      IF(Y.LT.1.)Y=1.
      CALL TRAA(X,Y,IPLU)
   44 IPLU=1
       CALL TRAA(XF,YF,0)
  193 CONTINUE
C    TRACE DE L AXE ET DES ABCISSES  (SIMPLIFIE)
      CALL TRAA(AXEX,HS4,0)
      CALL TRAA(-0.5,HS4,1)
      CALL TRAA(0.,HS4,1)
      H1=HS4+H4S5
      CALL TRAA(0.,H1,1)
      CALL TRAA(-0.5,H1,1)
```

```
      CALL NOMBA(-1.5,H1,0,1.0,1,0.2,0.3,1.,0.)
      CALL TRAA(0.,HS4,0)
      NREF=AXEX/10.
      NQS=0
      DO 36 N=1,NREF
      RN=N
      ABC=SIG1G+RN*DX10
      DO 37 NQ=1,9
      NQS=NQS+1
      X=NQS
      CALL TRAA(X,HS4,0)
   37 CALL TRAA(X,HS5M3,1)
      NQS=NQS+1
      X=NQS
      CALL TRAA(X,HS4,0)
      CALL TRAA(X,HS5M5,1)
   36 CALL NOMBA(X,HS5M10,0,ABC,2,0.2,0.2,1.,0.)
C     REINITIALISATION
  192 CALL PNUMA(AXEX+5.,0.,NADR,0.,0.)
C
      NADR=-900-IFICH
      CALL PNUMA(0.,0.,NADR,0.,0.)
      FI=KPLOT*(AXEX+5.)/100.
      PRINT 42,FI
   42 FORMAT(/,' TRACE TERMINE (',F9.2,' METRES.)')
C     REWIND 12
      RETURN
      END
```

```
C  *****    PROGRAMME  ''T102''    ************                    T103
C
C        CE PROGRAMME EXPLOITE LES NPRIM VALEURS DU SPECTRE CALCULE PAR TFID4.
C    NPRIM EST UNE PUISSANCE DE 2 LUE DANS LES CARTES DONNEES. LES NPRIM/2 PREMIE
C    RES VALEURS SUR  LE FICHIER DISQUE 1 ET LES NPRIM/2 DERNIERES SUR LE FICHIER
C    11.          IL NORMALISE LES VALEURS PUIS LES ECRIT PAR BLOCS DE 1024 VALEURS
C    SUR LA BANDE FT16. IL INTERPOLE  LES POINTS PRIMAIRES AINSI OBTENUS DANS LA
C    REGION OU S'EST PRODUITE LA NORMALISATION(SIG1).L'ECHELLE DX EST CALCULEE
C    AUTOMATIQUEMENT POUR QUE LE TRACE DE 1 OU 2 BLOCS AUTOUR DE SIG1 NE DEPASSE
C    PAS 2 OU 4 METRES DE PAPIER. LE PROG. SORT LE COEF. DE REDUCTION A APPLI-
C    QUER AUX VALEURS INTERPOLEES POUR QU'ELLES SOIENT NORMALISEES A 1.
C    M8=DIMENSION DU TABLEAU BB.       POUR LSIZE VOIR LE BLKSIZE DU DEFINE FILE.
C
      REAL*8 SIG1B,DELTAB,DELTAG,BENGTH
      REAL*8FRANGE,DX,DELTAD,ABC,DX10,RN
      REAL*8 RIFR,TRANCH,RMR,RNSB,DSIGMA,RNBLOC,SIGOG
      DIMENSION B(1050),AA(109),G(5120),BB(8192)
      DIMENSION CB(1024),P(5120),IBUF(1013)
      EQUIVALENCE (P(1),G(1)),(B(21),CB(1))
      DATA IPEN/1/,NADR/1/,IR/1/,LSIZE/1024/,M8/8192/,HT/25./,NBLOC/2/
C     FRANGE=+0.350798647D-05
      FRANGE=+0.791239320D-05                                    3/7/72 C
      DEFINE FILE 1(1024,1024,U,IV),11(1024,1024,U,IV)
      READ(5,2011)M,NPRIM,NU,IPLO,IFICH,INORM, IFR,SIGB,IAP
 2011 FORMAT(I10,5I5,  ,I5,D15.8,I1)
      NPRIM=NPRIM*1024
      IF(NPRIM.LT.32768)GO TO 6                                  ERREUR
      M81024=M8/1024
      ND1024=NPRIM/1024
      NSEGM=M8/LSIZE
      L28=NPRIM/(M8*2)
      PRINT 2,NPRIM,M,NU,M8,IPLO,IFICH,INORM,FRANGE,IFR,SIGB,IAP,NSEGM
    2 FORMAT(' NPRIM=',I7,' M=',I7,' NU=',I7,' M8=',I7,' IPLO=',I2,' IFI
     1CH=',I2,' INORM=',I2,/,' FRANGE=',D15.8,' IFR=',I5,' SIGB=',D15.8,
     2' IAP=',I1,' NSEGM=',I6)
      RIFR=IFR
      DELTAD=FRANGE*RIFR
      TRANCH=1./(2.*DELTAD)
      MR=SIGB*DELTAD
      RMR=MR
      DSIGB=SIGB-(RMR/DELTAD)
      IF(DSIGB.GT.TRANCH)IR=3
      RNSB=NPRIM
      DELTAB=1./(2.*DELTAD*RNSB)
      DELTAG=DELTAB/5.
      BENGTH=5120.*DELTAG
      SIG1B=RMR/DELTAD
      IF(IR.NE.1)SIG1B=SIG1B+TRANCH
      PRINT 13,DELTAD,MR,IR,SIG1B,DELTAB,BENGTH,DELTAG
   13 FORMAT(' DELTAD=',D15.8,' MR=',I5,' IR=',I5,' SIG1B=',D15.8,' DELT
     1AB=',D15.8,' BENGTH=',D10.3,' DELTAG=',D10.3)
C
C    CALCUL DE BMAX ENTRE B(500),ET B(NPRIM-500) OU B(N1) ET B(N2) SI INORM=1
      BMAX=0.
      I1=1
      L48=L28*2
      LI=L28*NSEGM
      L1=1
      L2=L48
      N1=500
      N2=NPRIM-500
      IA=500
      IB=M8-500
      IF(INORM.NE.1)GO TO 20
      READ(5,21)SIGN1,SIGN2
   21 FORMAT(2D15.8)
      PRINT 22,SIGN1,SIGN2
   22 FORMAT(' SIGN1=',D15.8,' SIGN2=',D15.8)
      N1=(SIGN1-SIG1B)/DELTAB
      L1=(N1/M8)+1
      IA=N1-(L1-1)*M8
      N2=(SIGN2-SIG1B)/DELTAB
      L2=(N2/M8)+1
      IB=N2-(L2-1)*M8
   20 DO 28 L=L1,L2
      MKP=(L-1)*NSEGM+1
      IF(L.LE.L28)GO TO 100
      MKP=MKP-LI
      I1=11
  100 READ(I1'MKP)BB
      II1=1
      II2=M8
      IF(L.EQ.L1)II1=IA
      IF(L.EQ.L2)II2=IB
      DO 28 I=II1,II2
      IF(BMAX-BB(I))29,28,28
   29 BMAX=BB(I)
      IBMAX=(L-1)*M8+I
   28 CONTINUE
      PRINT 23,IBMAX,BMAX,N1,N2
   23 FORMAT(' BMAX=B(',I7,')=',E12.5,' RECHERCHE ENTRE B(',I7,') ET B('
     1,I7,')')
C
C  *** NORMALISATION DES B SUR DISQUE
      PRINT 24
   24 FORMAT(/,' 10 PREMIERES VAL. DES BLOCS DE M8 VAL. NORMALISEES.')
      LMAX=0
      J1=1
      WRITE(16)NU,NU,NU,ND1024,SIG1B,DELTAB
 6515 DO 515 L9=1,L28
      LMAX=LMAX+1
      MKP=(L9-1)*NSEGM+1
      READ(J1'MKP)BB
      DO 516 I=1,M8
  516 BB(I)=BB(I)/BMAX
C     IF(LMAX.EQ.IMAX)PRINT15,BB
C  *** COPIE DES  B NORMALISES SUR LA BANDE 16
      I1=1
      I2=1024
      DO 517 I=1,M81024
      WRITE(16)(BB(J),J=I1,I2)
      I1=I1+1024
  517 I2=I2+1024
  515 PRINT 54,(BB(J),J=1,10)
```

```
   54 FORMAT(1X,10E12.2,' ***X***')
      IF(J1.EQ.11)GO TO 6525
      J1=11
      GO TO 6515
 6525 END FILE 16
      REWIND 16
      PRINT 205
  205 FORMAT(10X,'FIN DE LA COPIE DES B NORMALISES SUR LA C6',//)
C
C****       INTERPOLATION  ET  TRACE DU SPECTRE (SI IPLO=1)
C
      READ (16)NU,NU,NU,ND1024,SIG1B,DELTAB
C
C *** CALCUL DE LA FONCTION D INTERPOLATION AA.
C *** AA=TF. D UN RECT. DE LONG. 1 ECHANT. AVEC  201 VAL.(DELTAD=0.005)
C *** CE RECT. PEUT ETRE PONDERE SUR MP=( 201*M)/NPRIM VALEURS SI IAP=1
      MP=201
      DO 1 I=2,200
    1 B(I)=1.0
      B(201)=0.5
C   PONDERATION DE L'INTERF.
      IF(IAP.EQ.0)GO TO 12
      MP=(201*M)/NPRIM
      Q=(MP-1)*(MP-1)
      IF(IAP-2)1935,1936,1937
 1935 R=0.                                                        AP01
      S=0.
      T=1.
      W=0.
      V=2.
      GO TO 5000
 1936 R=0.06267                                                   AP02
      S=0.33213
      T=0.21839
      W=0.38680
      V=2.
      GO TO 5000
 1937 IF(IAP-3)1939,1939,1940
 1939 R=0.23978                                                   AP03
      S=0.45808
      T=0.22499
      W=0.077208
      V=2.
      GO TO 5000
 1940 R=0.                                                        AP04
      S=0.
      T=1.
      W=0.
      V=5./2.
 5000 DO 9 I=2,MP
      FI=I-1
      F=1.-(FI*FI)/Q
      PK=R+S*F+T*F**V+W*F***3
    9 B(I)=B(I)*PK
C *** CALCUL TF CLASSIQUE  AA=TF.(B)
C *** CALCUL TF CLASSIQUE  AA=TF.(B)
   12 AMAX=0.5
      DO 4 IK=2,MP
    4 AMAX=AMAX+B(IK)
      SIA=+0.3141592653D-02
      DO 7 IJ=56,105
      FIJ=IJ-55
      SIG=SIA*FIJ
      AA(IJ)=0.5
      DO 3 IK=2,MP
      FIK=IK-1
    3 AA(IJ)=AA(IJ)+B(IK)*COS(SIG*FIK)
C *** NORMALISATION ET PONDERATION DE LA FONCTION AA
    7 AA(IJ)=(AA(IJ)/AMAX)*(1.-(FIJ*FIJ)/2500.)**2
      AA(55)=1.
      DO 8 I=106,109
    8 AA(I)=0.
C *** DEFINITION PAR SYMETRIE DE LA FCT. D'INTERP. AA COMPLETE
      DO 181 I=1,54
  181 AA(I)=AA(110-I)
      PRINT 183,MP
  183 FORMAT(' FCT. D INTERPOL. AA AVEC MP=',I5,/)
C
C ***CARACTERISTIQUES DU TRACE QUI DOIT ETRE FAIT DU BLOC NB1 OU SE TROUVE
C *** SIG1 A CELUI OU SE TROUVE SIG2. LE TRACE EST DECOUPE EN NPLOT PLOTS
C *** DE NBLOC BLOCS
      RBMAX=IBMAX-10
      SIG1=RBMAX*DELTAB+SIG1B
      SIG2=SIG1+20.*DELTAB
      DX=10.*DELTAB
      PRINT 185,SIG1,DX
  185 FORMAT(' TRACE A FAIRE VERS ',F7.0,' AVEC DX=',F12.3,' CM-1/CM')
      GMAX=0.0
      HS4=HT/13.5
      H4S5=HT/1.35
      H4SN=H4S5/2.
      HS5M3=HS4-0.2
      HS5M5=HS4-0.4
      HS5M10=HS4-0.8
      NB1=(SIG1-SIG1B)/BENGTH+1.
      NB2=(SIG2-SIG1B)/BENGTH+1.
      IF(NB1.LT.1)NB1=1
      IF(NB2.GT.ND1024)NB2=ND1024
C *** ON DOIT TRACER DU BLOC NB1 AU BLOC NB2
      PRINT 14,NB1,NB2
   14 FORMAT(' NB1=',I10,' NB2=',I10)
      IF(NB1.EQ.1) GO TO 187
      NB1M1=NB1-1
      DO 188 I=1,NB1M1
  188 READ(16)CB
      DO 189 I=1025,1044
      J=I-1024
  189 B(J)=B(I)
      GO TO 190
  187 DO 191 I=1,20
  191 B(I)=0.0
  190 RNB1M1=NB1-1
C
      NBLOC=NB2-NB1+1
      PRINT 207,NBLOC
  207 FORMAT(' NBLOC IMPOSE PAR LE PG.',I8)
      RNBLOC=NBLOC
      DSIGMA=RNBLOC*BENGTH
      SIGOG=SIG1B+RNB1M1*BENGTH-10.*DELTAB
      IF(IPLO.NE.1)GO TO 200
      CALL IBENA(IBUF,1013,14)
      CALL PNUMA(0.,0.,NADR,0.,0.)
      FU=NU
      CALL NOMBA(5.,HT,0,FU,0,0.2,0.2,1.,0.)
       CALL TRAA(0.,0.,0)
  200 KKK=0
C
      DO 193 KB=1,NBLOC
      READ(16)CB
C
C     CONVOLUTION
      DO 194 K=1,5120
      G(K)=0.
      NZ=(K-1)/5
      KZ=NZ*5-K+1
      DO 194 N=1,21
      NZ=NZ+1
      KZ=KZ+5
  194 G(K)=G(K)+B(NZ)*AA(KZ)
      DO 196 I=1,20
      J=I+1024
  196 B(I)=B(J)
C
C     CALCUL DE GMAX
      DO 300 K=1,5120
  300 IF(GMAX.LT.G(K))GMAX=G(K)
      IF(IPLO.NE.1)GO TO 193
C
```

```
C     TRACE D UN BLOC DE 5120 VALEURS INTERPOLEES
      DO 35 J=1,5120
      FKKK=KKK
      X=(FKKK*DELTAG)/DX
      KKK=KKK+1
      Y=P(J)*H4SN+HS4
      IF(Y.GT.30.)Y=30.
      IF(Y.LT.0.)Y=0.
   35 CALL TRAA(X,Y,IPEN)
  193 CONTINUE
      IF(IPLO.NE.1)GO TO 204
C     TRACE DE L AXE ET DES ABCISSES. (SIMPLIFIE)
      AXEX=DSIGMA/DX
      DX10=DX*10.
      CALL TRAA(AXEX,HS4,0)
      CALL TRAA(-0.5,HS4,1)
      CALL TRAA(0.,HS4,1)
      H1=HS4+H4S5
      CALL TRAA(0.,H1,1)
      CALL TRAA(-0.5,H1,1)
      CALL NOMBA(-1.5,H1,0,2.0,1,0.2,0.2,1.,0.)
      NREF=AXEX/10.
      NQS=0
      DO 36 N=1,NREF
      RN=N
      ABC=SIGOG+RN*DX10
      DO 37 L=1,9
      NQS=NQS+1
      X=NQS
      CALL TRAA(X,HS4,0)
   37 CALL TRAA(X,HS5M3,1)
      NQS=NQS+1
      X=NQS
      CALL TRAA(X,HS4,0)
      CALL TRAA(X,HS5M5,1)
   36 CALL NOMBA(X,HS5M10,0,ABC,2,0.2,0.2,1.,0.)
C
      NADR=-900-IFICH
      CALL PNUMA(AXEX+6.,0.,NADR,0.,0.)
      FI=AXEX/100.
      PRINT 42,FI
   42 FORMAT(' TRACE TERMINE.  (',F7.2,' METRES.)')
  204 PRINT 301,GMAX
  301 FORMAT(/,'   POUR NORMALISER LES VALEURS INTERPOLEES DU SPECTRE,DI
     1VISER TOUTES LES VALEURS PAR GMAX='F7.4)
    6 STOP
      END
```

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

CENTRE INTER-REGIONAL DE CALCUL ELECTRONIQUE

Le programme de pointés automatique de raies : "DERPTE"

par H. DELOUIS

Version 1.4
Janvier 1973

TABLE DES MATIERES

I. - INTRODUCTION - GENERALITES. pages

1. - Ce que fait le programme DERPTE ? 1

2. - Principe de la détection automatique des raies et différentes versions de ce programme. 2

II. - METHODES ADOPTEES POUR L'ETUDE D'UNE RAIE. (sous-programme "POINTG")

1. - Etude de la position. 4

2. - Etude de la largeur équivalente. 7

3. - Organisation du programme "DERPTE". 10

4. - L'algorithme de détection automatique des raies et son adaptation au cas des raies de forme complexe. 13

5. - Test de précision. 17

III. - CARTES DE CONTROLE - CARTES DONNEES.

1. - QUICK - DERPTE. 19

2. - DERPTE. 21

3. - Exemples de durée d'exécutions sur 360/75 IBM. 23

IV. - LES FORMATS DE SORTIE DE CE PROGRAMME.

1. - Liste des résultats.

- a) "ILIST=0" ou "QUICK-DERPTE" 24
- b) "ILIST=1" 24
- c) La carte "format" 25

2. - Tracé éventuel du spectre, et de sa dérivée seconde. 28

3. - Sauvegarde des résultats sur bande magnétique "FT17" 32

4. - Remarque à propos des intensités des raies. 32

V. - LISTE DU PROGRAMME DERPTE. 33

LE PROGRAMME DERPTE

- Raies d'émission ou d'absorption des spectres à large étendue spectrale.

- Recherche automatique, puis calcul de leur position précise, de leur largeur équivalente et de leur surface.

CHAPITRE 1

1. - Ce que fait le programme DERPTE ?

Commençons par passer en revue les différentes tâches effectuées par ce programme.

- a) Il s'applique à des spectres déjà calculés et souvent déjà tracés, tout au moins partiellement. Le nombre de points primaires de ces spectres est compris entre 50.000 et 2.10^6 avant interpolation. Ces points sont stockés sur une bande magnétique "FT16" (9 pistes, 1600 bpi) qui est conservée et cataloguée. Ils y sont groupés par blocs de 1024 valeurs.

- b) Le programme commence par lire ces points primaires du spectre sur la bande "FT16". Il les interpole, pour avoir un échantillonnage cinq fois plus serré.

Cette interpolation est limitée au domaine spectral compris entre deux nombres d'onde "SIG1" et "SIG2" choisis par l'utilisateur. La recherche des raies sera donc elle aussi limitée à ce domaine. On profite de cette interpolation pour apodiser éventuellement.

- c) Après avoir interpolé un bloc de 1024 valeurs primaires, le programme fait successivement :

  - La recherche automatique des raies qui se trouvent dans ce bloc.
  - Le calcul de leur intensité.
  - Le calcul de leur position précise (en double précision jusqu'à 10^{-10} près si la résolution et le rapport signal/bruit le permettent.
  - Le calcul d'un critère de symétrie.
  - Le calcul de sa surface et de sa largeur équivalente.

- d) Chaque raie trouvée et mesurée, est repérée par un numéro sur la liste. Le programme peut sur option tracer le domaine spectral analysé (SIG1, SIG2) et y repérer par leurs numéros les raies qu'il y a détectées. Le tracé est alors enregistré sur bande magnétique "FT14" (9 pistes, 800 bpi).

- e) Le numéro, la position, l'intensité et le critère de symétrie de chaque raie peuvent être conservés sur une bande magnétique "FT17" (9 pistes, 1600 bpi).

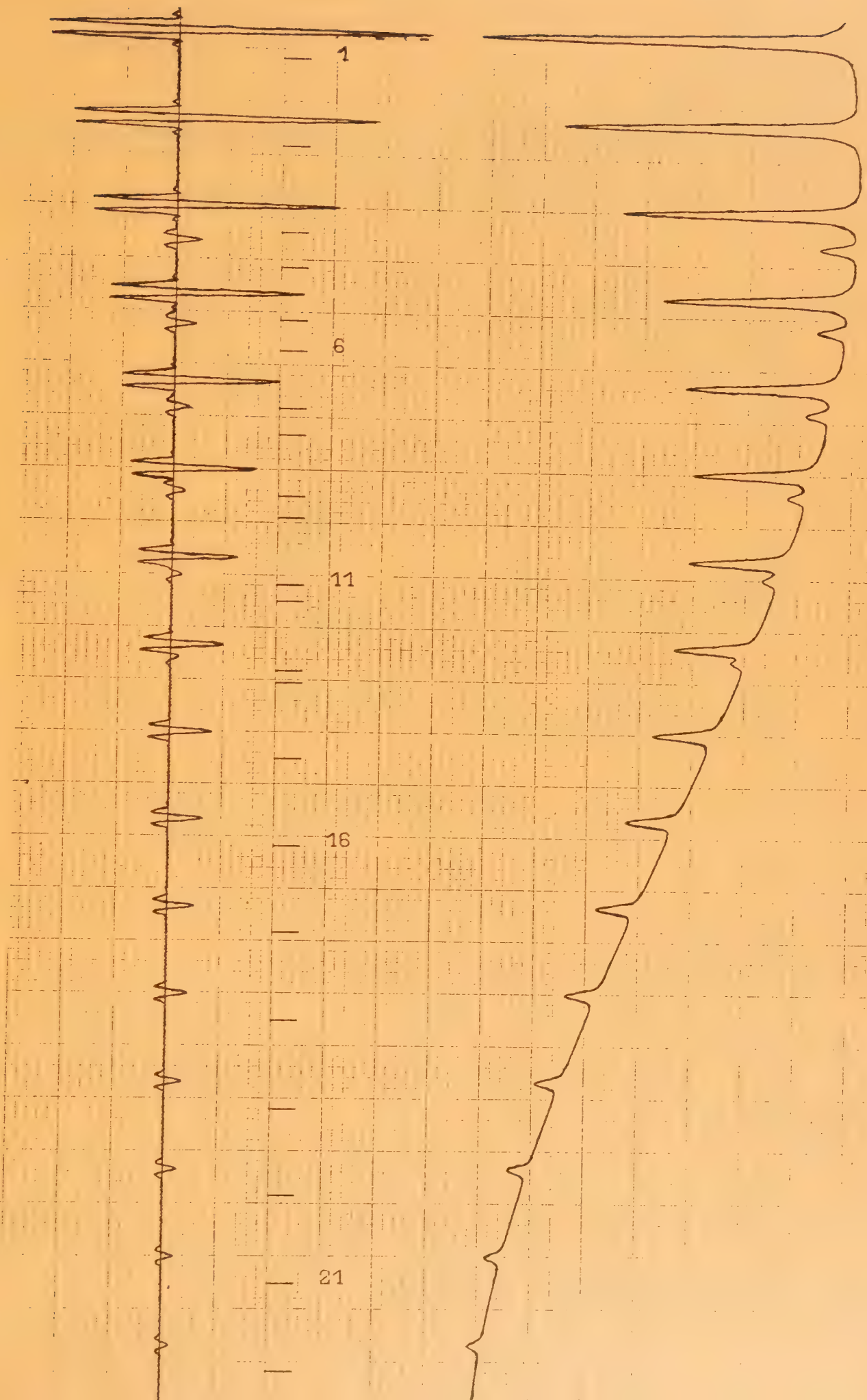
Nous verrons dans le chapitre IV, les différentes possibilités de liste et d'exploitation de tous ces résultats.

2. - Principe de la détection automatique des raies. Différentes versions de ce programme.

La recherche automatique des raies, est faite en repérant toutes celles pour lesquelles la dérivée seconde du spectre dépasse un niveau constant "HNIV" fixé par l'utilisateur dans les cartes données. L'intérêt de la dérivée seconde, réside dans le fait qu'elle permet d'éliminer le fond continu variable du spectre comme le montre bien la figure p.3.

La seule difficulté réside dans le choix du niveau de détection "HNIV". Ce choix dépend de la résolution, du rapport s/b. et du type de raies que l'on cherche à détecter. Cette détermination ne peut donc être faite que de façon expérimentale pour chaque spectre à étudier. Pour la faciliter, un programme spécial appelé "QUICK-DERPTE" a été écrit. Ses possibilités ont été volontairement limitées pour qu'il soit plus facile à utiliser et qu'il passe rapidement en machine. En particulier, il arrête son travail de détection dès qu'il a trouvé 50 raies (dans l'hypothèse où il y aurait détection de plus de 50 raies dans le domaine SIG1, SIG2) et il se contente de lister le numéro, la position et l'intensité de ces raies. Ses caractéristiques sont les suivantes : Temps d'exécution de l'ordre de 4 secondes sur 360/75 IBM, encombrement en mémoire centrale de 115 K octets, édition de moins de 300 lignes, un seul montage de bande ("FT16"). Ce programme "QUICK-DERPTE" permet donc en faisant trois ou quatre essais rapides suivis de confrontation avec le tracé correspondant de fixer la valeur de "HNIV" qui sera utilisée dans le programme "DERPTE" normal.

Le programme "DERPTE" normal peut détecter un nombre illimité de raies, et son encombrement en mémoire centrale est de 144 K octets. On pourra limiter le nombre de raies à étudier en limitant le nombre de lignes en édition. Il utilise un fichier sur bande ("FT16") et deux autres fichiers sur options. Il s'agit des fichiers sur bandes "FT14" pour le tracé et "FT17" pour la conservation des résultats.



TRACE D'UN SPECTRE D'ABSORPTION ET DE LA DERIVEE SECONDE DE CE SPECTRE

CHAPITRE II

((*)) Méthodes adaptées pour l'étude d'une raie. Mise en oeuvre par sous programme "POINTG" inclus dans le programme "DERPTE"

Il s'agit d'un programme qui, à partir des valeurs échantillonnées G(I) d'un spectre, permet l'étude rapide de la position, de la forme et de la largeur équivalente de raies. Les valeurs échantillonnées du spectre se trouvent sur bande magnétique et sont accompagnées de trois caractéristiques; le nombre total de valeurs N_g, l'abcisse du premier point de l'échantillonnage soit σ_1 et enfin le pas de l'échantillonnage, soit $\Delta\sigma$.

L'étude systématique des raies dans les bandes par ordinateur est devenue indispensable, tant au point de vue de la précision que du gain de temps depuis que le calcul par T.F. de spectres à haute résolution est devenu une réalité. (10^7 valeurs par spectre).

Dans une première partie, nous montrerons de quelle façon le programme détermine l'abcisse et la forme d'une raie.

Dans une deuxième partie, nous étudierons le problème de la détermination de la surface et de la largeur équivalente de cette même raie.

Enfin, dans une troisième partie, nous donnerons quelques détails sur l'organisation du programme.

- 1) Etude de la position d'une raie.

Chaque raie est caractérisée par la position approximative de son sommet repérée sur le tracé du spectre, soit σ_s, et par sa largeur approximative $\Delta\sigma_2$. Ce sont là les données de départ qui sont maintenant calculées automatiquement par le programme.

La valeur σ_s permet de trouver l'emplacement de la raie à étudier parmi les valeurs échantillonnées G(k) du spectre. L'étude de cette raie va se faire sur un domaine qui va de $\sigma_s - \Delta\sigma_2$ à $\sigma_s + \Delta\sigma_2$. Soient $G(N_p)$, $G(N_{p+1})$, ... $G(N_d)$ les valeurs correspondant aux abcisses $\sigma(N_p)$,..., $\sigma(k)$..., $\sigma(N_d)$ qui se trouvent à l'intérieur de ce domaine.

../..

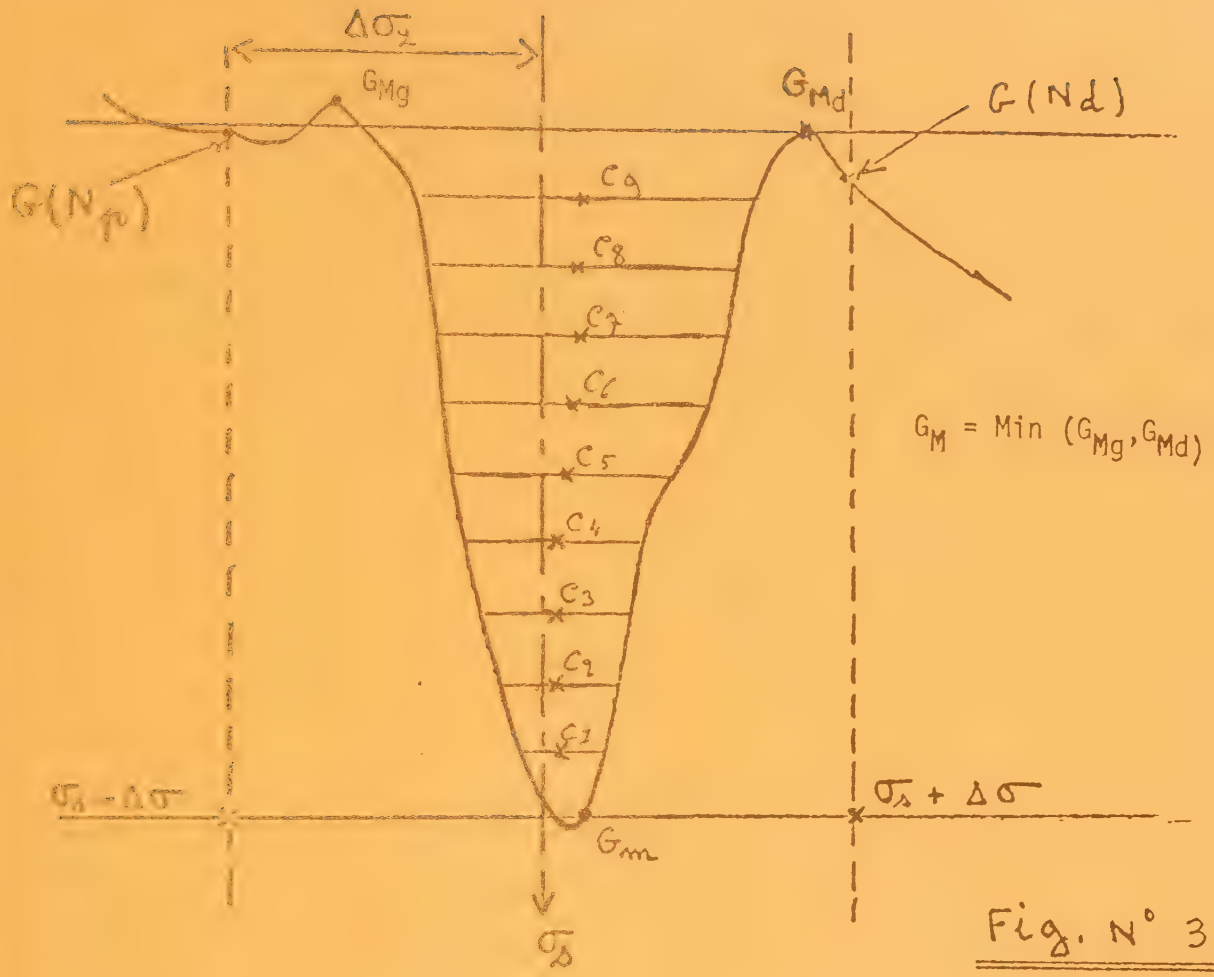


Fig. N° 35

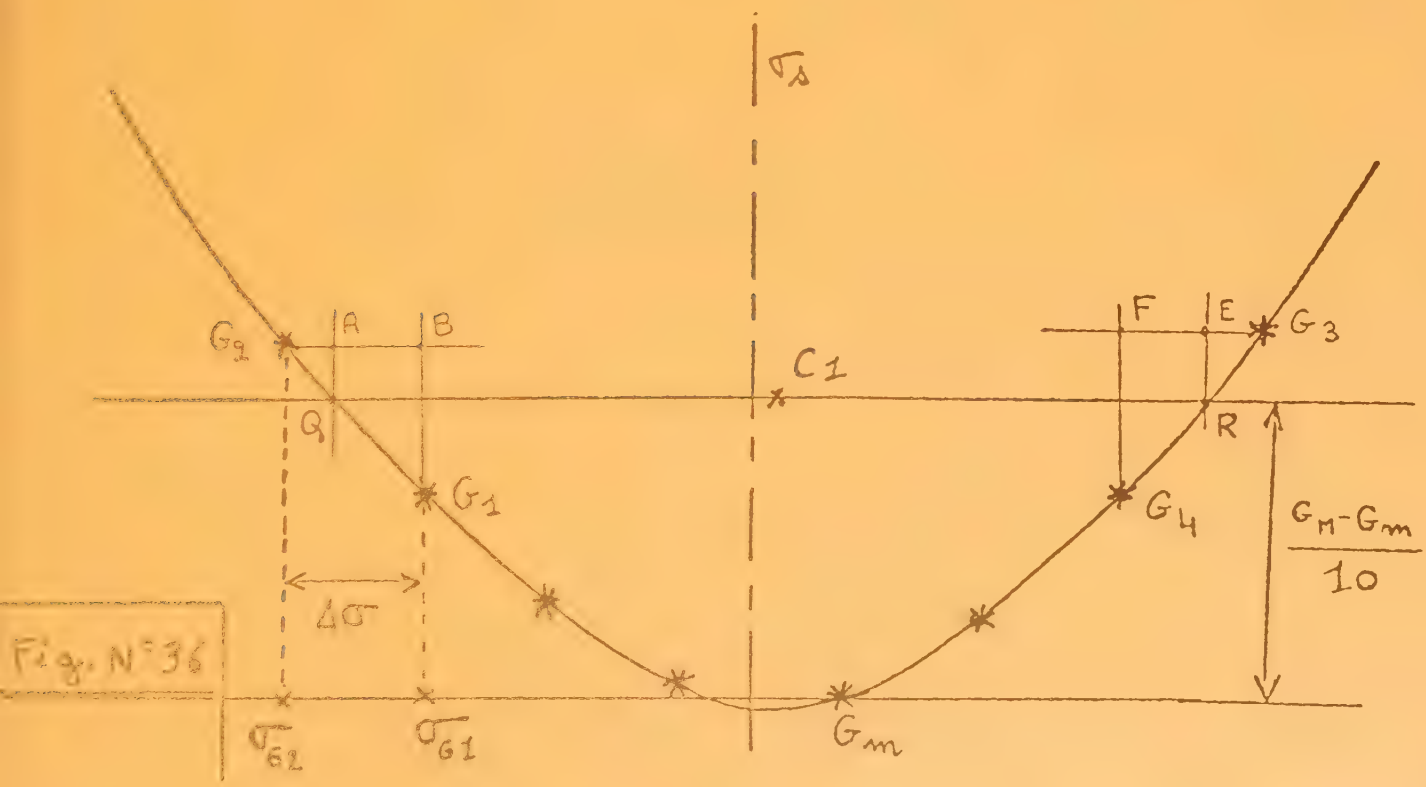


Fig. N° 36

$$\sigma(1) = \sigma_1$$
$$\sigma(k) = \sigma_1 + (k-1)\ \Delta\sigma$$
$$\sigma(N_p) = \sigma_1 + (N_p-1).\Delta\sigma \quad \Big)$$
$$\sigma(N_d) = \sigma_1 + (N_d-1).\Delta\sigma \quad \Big) \quad \text{raie étudiée}$$

Parmi les valeurs G(k) comprises entre $G(N_p)$ et $G(N_d)$ on recherche alors la plus grande((*)), soit $\underline{G}_M$, et la plus petite, soit $\underline{G}_m$, puis on définit 9 niveaux équidistants H_1, H_2,....,H_9 pour lesquels on définit le milieu des cordes correspondantes, soit :

C_1, C_2,..., C_9 (voir figure 35)

$$H_k = G_m + k.\ \frac{G_M - G_m}{10}$$

Etude de C_1, C_2,....,C_9.

Supposons par exemple qu'il s'agisse de déterminer l'abcisse σ_{c1} du point C_1, le centre de la corde qui traverse la raie au niveau H_1. En se reportant à la figure 3 sur laquelle les valeurs G(k) sont repérées par des points le long du profil de la raie, on voit que la première chose à faire est de déterminer les couples de valeurs (G_1, G_2), (G_3, G_4) qui encadrent sur le flanc gauche, puis sur le flanc droit de la raie le niveau H_1. Cette recherche se fait en comparant les valeurs G(k) au niveau H_1, en supposant, bien sûr, que le profil du spectre à l'intérieur du domaine $\sigma_s - \Delta\sigma_2$, $\sigma_s + \Delta\sigma_2$ ne traverse que deux fois le niveau. Il faut alors déterminer l'abcisse des points Q et R en supposant que le profil de la raie entre les points G_1 et G_2 puis G_3 et G_4 est une portion de droite. En se reportant aux notations de la fig. 36 nous avons alors :

$$\sigma_Q = \sigma_{G2} + G_2 A$$

or $G_2A = \Delta\sigma\ .\ \frac{AQ}{BG_1} = \Delta\sigma\ .\ \frac{G_2 - H_1}{G_2 - G_1}$

$$\sigma_Q = \sigma_{G2} + \Delta\sigma.\ \frac{G_2 - H_1}{G_2 - H_1}$$

de même : $$\sigma_R = \sigma_{G3} - \Delta\sigma.\ \frac{G_3 - H_1}{G_3 - G_4}$$

((*)) $\underline{G_M}$: Entre $G(N_p)$ et $G(N_d)$ on cherche G_m puis de part et d'autre (voir figure 35) on recherche G_{Mg} le plus grand échantillon à gauche de G_m puis G_{Md} à droite. G_M es alors égal au plus petit des deux.

Nous avons alors : $\sigma_{c1} = \frac{\sigma_Q + \sigma_R}{2}$

et on refait la même chose pour les autres niveaux.

La valeur σ_{c1} est prise comme position de la raie, et on calcule ensuite les 8 différences de nombres d'onde $\sigma_{c2} - \sigma_{c1}$, $\sigma_{c3} - \sigma_{c1}$, ..., $\sigma_{c9} - \sigma_{c1}$ Ces différences donnent des renseignements sur le caractère plus ou moins symétrique de la raie. Si, par exemple, elles sont toutes systématiquement positives, c'est que le nombre d'onde du centre de gravité de la raie est plus grand que celui de la pointe.

Remarque.

Dans le cas où le spectre à étudier est de nature astronomique, il peut être nécessaire d'appliquer un certain nombre de corrections aux nombres d'ondes σ. Il y a d'abord les corrections de nature expérimentale proprement dites, telles que la correction de vide et la correction qui tient compte de la largeur du trou d'entrée de l'interféromètre. Ces deux corrections peuvent être appliquées en modifiant simplement la longueur d'onde étalon du système de référence qui permet l'échantillonnage de l'interférogramme. Il y a aussi les corrections d'effet Doppler qui doivent être appliquées uniquement aux nombres d'ondes des bandes d'origine solaire ou planétaire. Pour effectuer ces corrections avec le programme, l'expérimentateur fournit un coefficient k2, valable dans un petit domaine de longueur d'onde, et les nombres d'ondes correspondants dans le vide sont calculés de la façon suivante :

$$\sigma_{corrigé} = \sigma + \sigma . k_2$$

- 2) Calcul de la largeur équivalente.

Il s'agit en fait de déterminer trois grandeurs, le fond moyen de la raie, sa surface, sa largeur équivalente. La position de la raie est repérée par le nombre d'onde σ_{c1} calculé précédemment, et le programmeur fixe le domaine $\Delta\sigma_3$ sur lequel l'étude doit être faite.

Fond moyen.

Parmi les valeurs G(Nps), $G(N_{ps}+1)$,... $G(N_{ds})$ qui correspondent aux nombres d'ondes $\sigma(N_{ps})$, ..., $\sigma(N_{ds})$ avec :

$$\sigma_{c1} - \Delta\sigma_3 \leqslant \sigma\ (N_{ps}) \leqslant \sigma_{c1} - \Delta\sigma_3 + \Delta\sigma$$

$$\sigma_{c1} + \Delta\sigma_3 - \Delta\sigma \leqslant \sigma(N_{ds}) \leqslant \sigma_{c1} + \Delta\sigma_3$$

../..

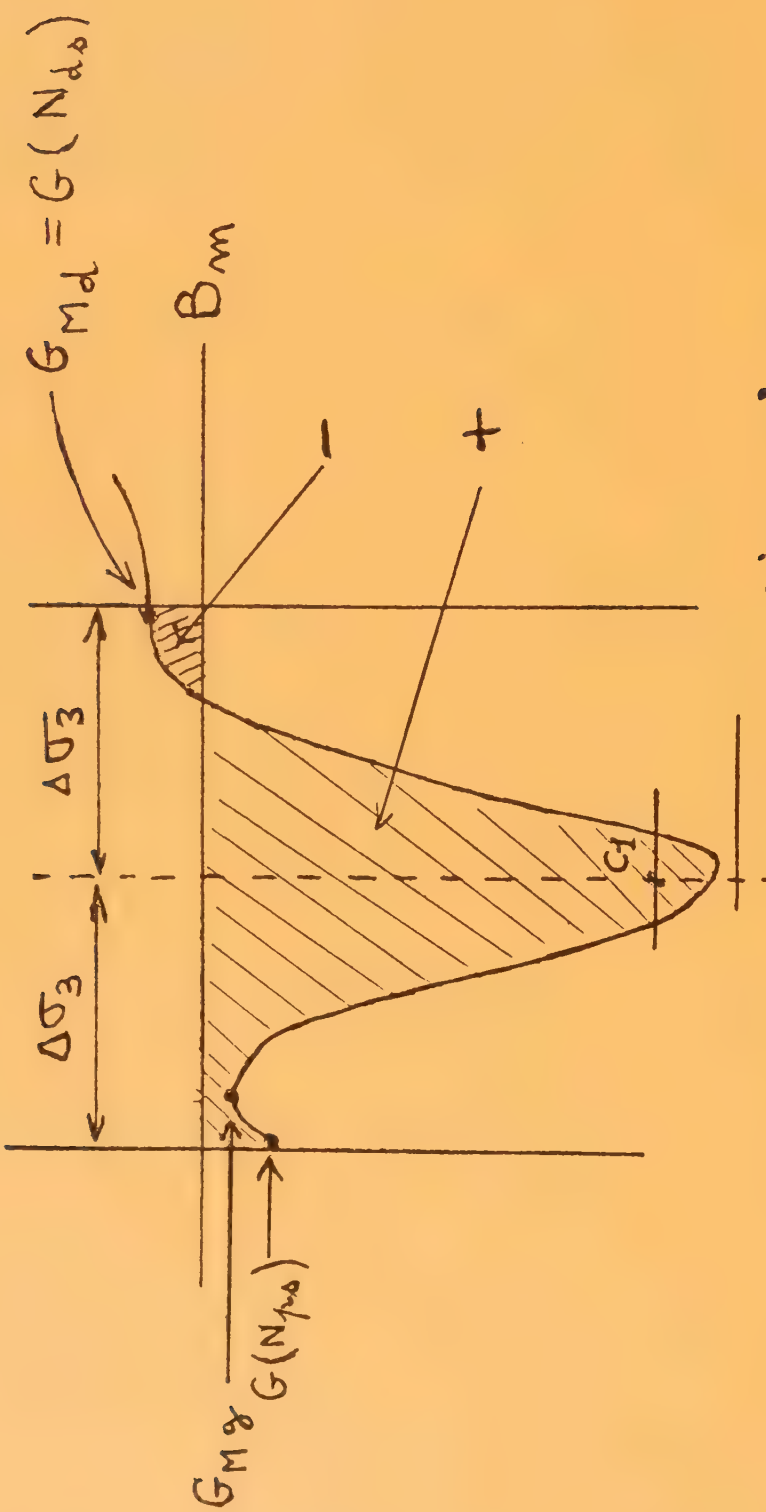
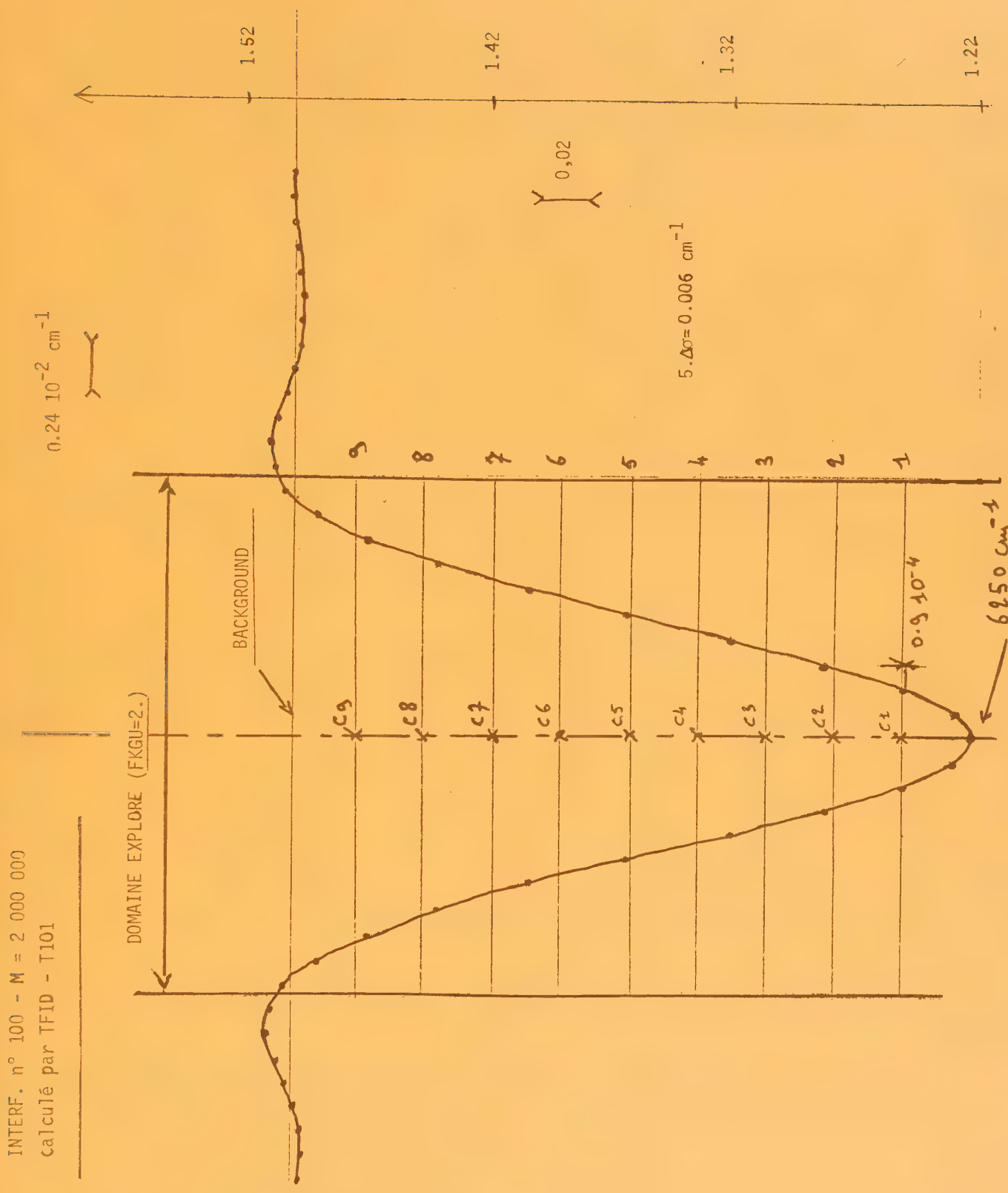


FIG. Nº 37



Relevé sur une raie synthétique des points après interpolation et apodisation. Disposition des éléments de mesure tels qu'ils ont été calculés par le programme DERPTE.

On recherche la plus grande valeur G_{Mg} à gauche, puis la plus grande valeur G_{Md} à droite du pic de la raie. (voir figure N° 37).

Le fond moyen de la raie, soit B_m est défini par :

$$B_m = \frac{G_{Mg} + G_{Md}}{2}$$

On définit alors les valeurs G'(k) des points du profil de la raie par rapport au niveau moyen B_m soit :

$$G'(k) = B_m - G(k)$$

La surface S de la raie est définie par l'expression suivante :

$$S = \Delta\sigma \left[G'(N_{ps})/2 + G'(N_{ps}+1) + \ldots + G'(N_{ds}-1) + G'(N_{ds})/2 \right]$$

La largeur équivalente L_{eq} est alors définie comme étant la largeur du rectangle ayant la même surface S que la raie, mais une hauteur égale au fond moyen de la raie, soit B_m d'où :

$$\boxed{L_{eq} = \frac{S}{B_m}}$$

3) Organisation du programme "DERPTE"

(Entre les apostrophes, on retrouve les symboles utilisés dans le programme).

Le programme lit les valeurs primaires B du spectre par blocs de 1024 valeurs sur la bande magnétique FT16. Comme il y a en général un grand nombre de ces blocs à étudier (du bloc "NB1" au bloc "NB2") et que la place en mémoire centrale est limitée, il va, à chaque fois qu'il aura lu un de ces blocs, le traiter complètement avant de passer au bloc suivant. Pour ce faire, chaque bloc lu est mis en mémoire centrale dans un tableau "B" de 1024 valeurs. Il est alors interpolé et les 5120 valeurs G ainsi obtenues sont stockées dans un tableau "H". Pour assurer la liaison entre chacun des blocs primaires ainsi étudiés, on fait précéder ces 5120 valeurs des 1024 dernières valeurs interpolées provenant du bloc précédent. (S'il

- ORGANISATION DU PROGRAMME DERPTE -

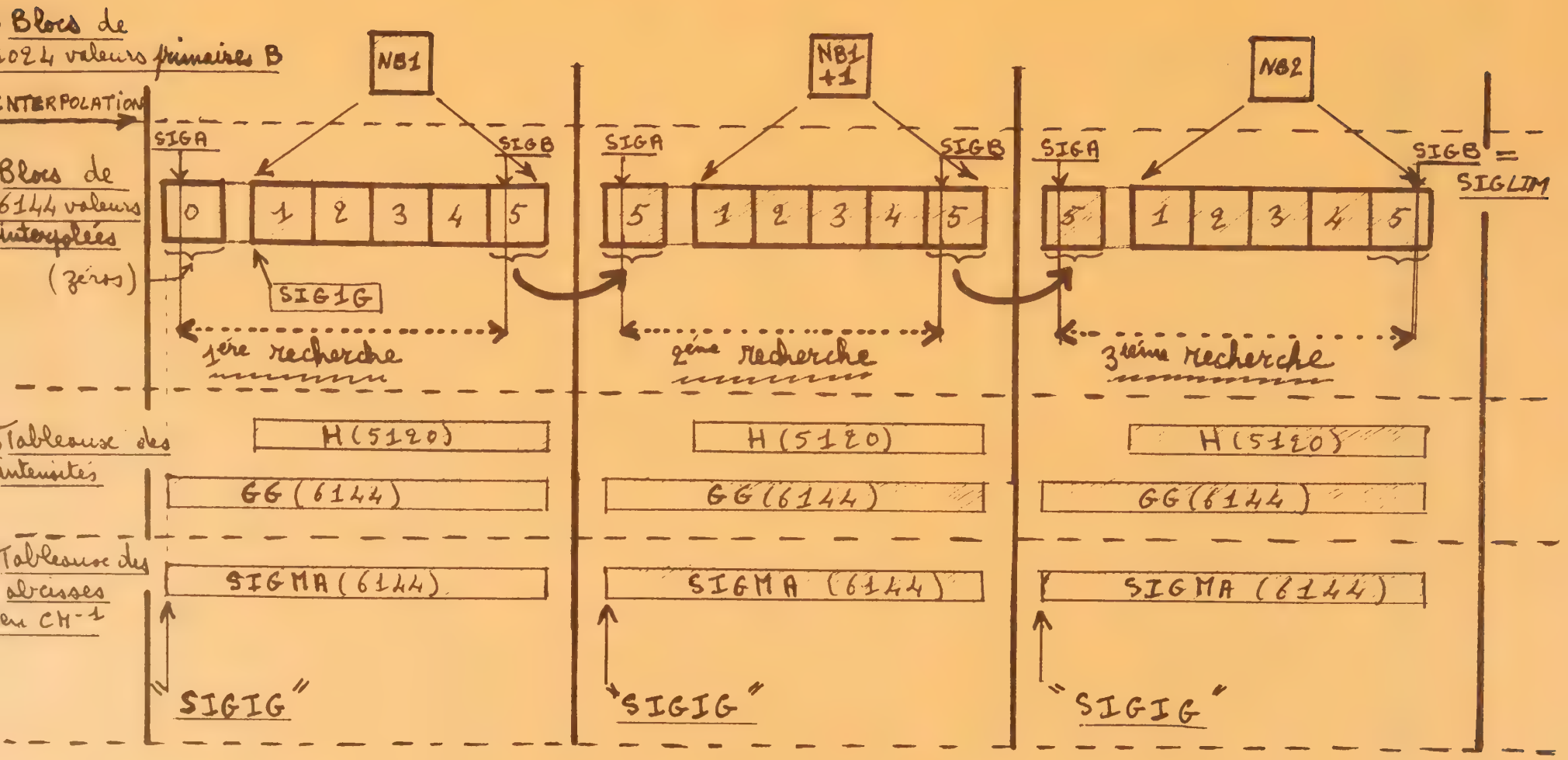


FIGURE - 38 -

La première raie à étudier se trouve dans le "NB1"^{ième} bloc de 1024 valeurs primaires (schématisé par un carré) et la dernière dans le "NB2"^{ième} bloc. La recherche et l'étude des raies se fait bloc par bloc.

s'agit du 1er bloc, on le fait précéder de zéros). On trouve donc simultanément en mémoire centrale, les 6144 valeurs G interpolées (dans le tableau "GG" lui même en "équivalence" sur "H") et les 6144 abscisses correspondantes en CM^{-1} (tableau SIGMA).

Le programme recherche alors les "NRAIE" raies se trouvant dans ce bloc en utilisant la dérivée seconde des valeurs G. Les abscisses des valeurs G sont comprises à chaque fois entre "SIGIG" et "SIGIG+1024*DELTAB" (DELTAB=pas d'échantillonnage primaire), mais pour éviter d'avoir à étudier des raies trop proches de ces deux limites, il élimine toutes celles qui ne sont pas comprises entre "SIGA" et "SIGB" (voir figure 38). Celles qu'il détecte avant "SIGA" sont sensées avoir été traitées avec le bloc précédent et celles après "SIGB" le seront avec le bloc suivant. La figure 38 illustre cette organisation dans le cas où il y a 3 blocs primaires à étudier (du bloc "NB1" au bloc "NB2"). Elle montre bien sur quels domaines successifs (1er, 2ème, 3ème recherche) les raies seront recherchées puis étudiées.

- Notations.

Les notations de la partie gauche sont celles du programme et celles de la partie droite correspondante sont celles utilisées dans cet exposé

NSG →	N_σ	AB →	G_2 A
SIG1G →	σ_1	ØB →	G_3 E
DELTAG →	$\Delta\sigma$	SIGG5 →	σ_Q
NNP →	N_p	SIGG6 →	σ_R
NND →	N_d	CØEF1 →	k_1
SIGGS →	σ_s	NNPS →	N_{ps}
DSIGS2 →	$\Delta\sigma_2$	BACKGR →	B_m
GGMIN →	G_m	SURF →	S
GGMAX →	G_M	ARGEQ →	L_{eq}
HAUT(I) →	H_i	GGMAXG →	G_{Mg}
SIGS(I) →	σ_{ci}	GGMAXD →	G_{Md}

- REMARQUE : Dans ce programme, tout ce qui concerne le calcul des abcisses (variables, tableaux et expressions) est en double précision.

4) L'algorithme de détection des raies et le cas des raies de forme complexe.

Reportons nous à la figure 5 de la page 28. Nous y voyons tracés une raie, sa dérivée seconde et le niveau C de détection. Si on se contente de détecter une raie lorsqu'un maximum de sa dérivée seconde dépasse le niveau HNIV et si on attend que cette même dérivée seconde redescende au dessous du niveau HNIV pour commencer la recherche d'une autre raie nous voyons qu'il risque de se poser un certain nombre de problèmes.

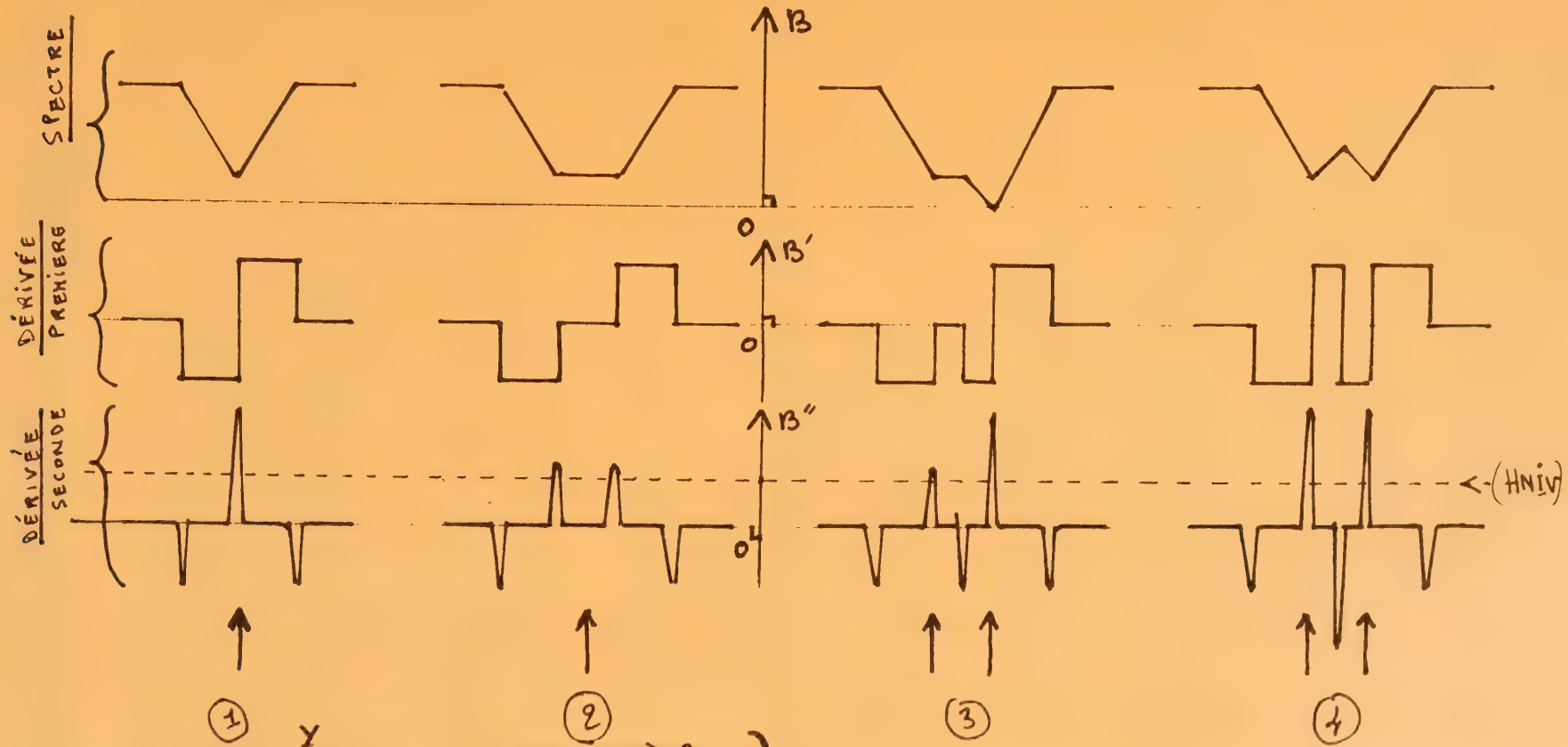
Quatre flèches A, B, C, D figurent, différents niveaux possibles pour HNIV. Voyons ce qui va se passer dans chacun de ces cas.

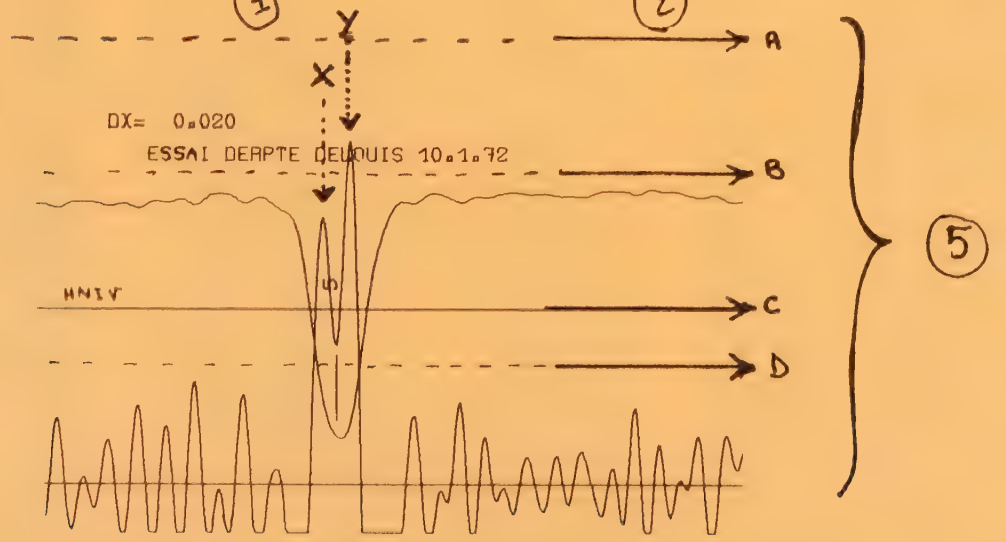
La dérivée seconde de cette raie de forme complexe présente la particularité d'avoir deux pics positifs X et Y de hauteurs différentes.

- HNIV en A ⟶ aucune raie détectée.
- HNIV en B ⟶ une raie détectée à l'abscisse de Y.
- HNIV en C ⟶ deux raies détectées en X et Y.
- HNIV en D ⟶ une raie détectée à l'abscisse de X.

- Suivant la valeur affectée à HNIV, on peut ainsi obtenir des résultats en contradiction apparente.

- L'étude de la forme de la dérivée seconde des raies schématiquement représentées sur les figures 1, 2, 3 et 4 de la page 28 nous a permis de définir un algorithme de détection un peu plus adapté.





Détection d'une raie de forme complexe en fonction du niveau HNIV.

Cet algorithme repose sur le principe arbitraire suivant : une raie est détectée dès qu'un maximum de la dérivée seconde du spectre dépasse le niveau HNIV mais au lieu d'attendre que cette dérivée redescende au dessous du niveau HNIV pour détecter une "nouvelle" raie, il continue la recherche des autres maxima éventuels de la dérivée seconde tant que cette dernière n'est pas redevenue négative. Deux cas alors se présentent :

a - Il trouve un seul maximum. Son abscisse est alors pris comme position de la raie détectée.

b - Il trouve plusieurs maxima. L'abscisse de la raie détectée est alors la moyenne de l'abscisse du maximum le plus à gauche et de celle du plus à droite.

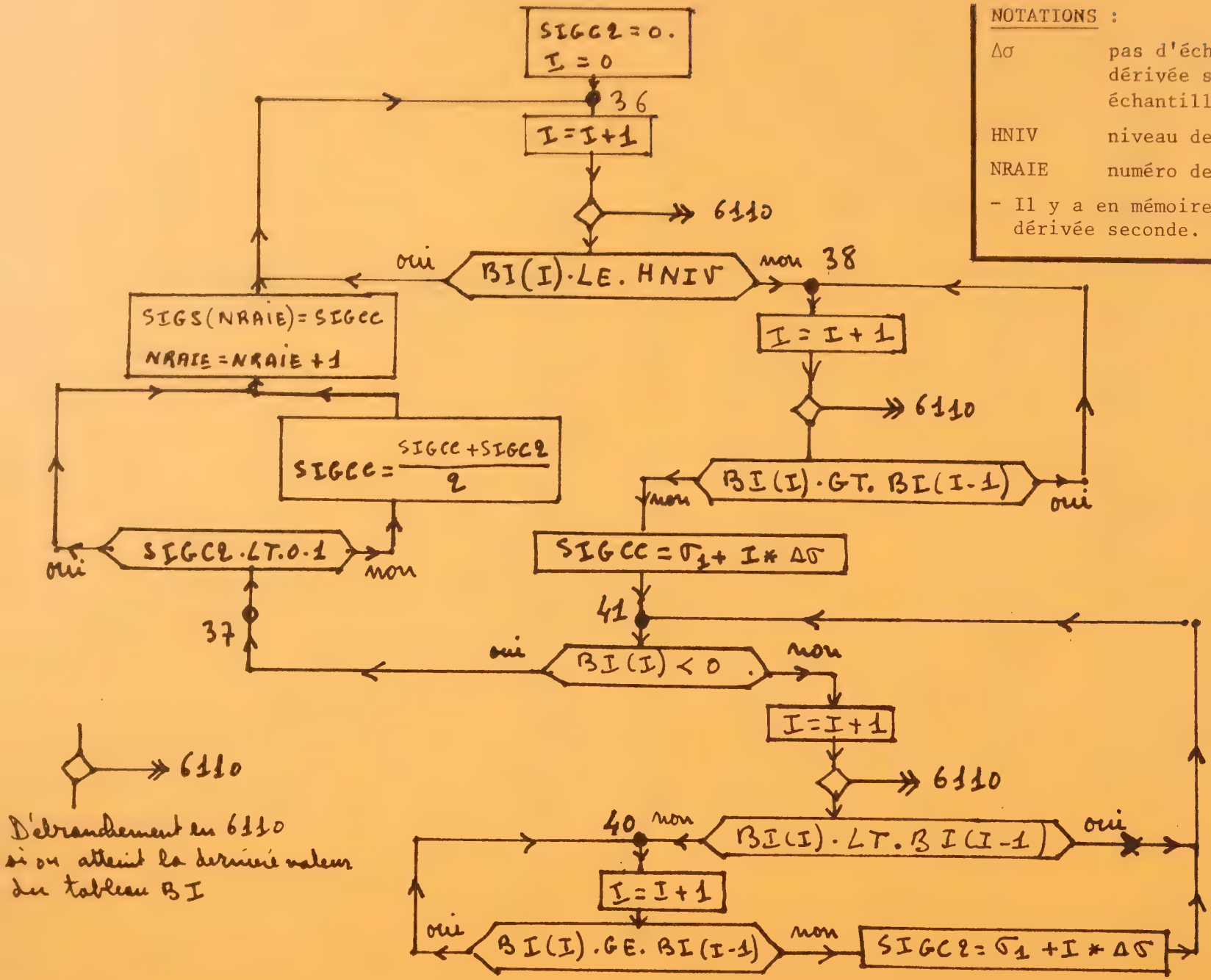
Les flèches sous les figures 1, 2, 3 et 4 montrent alors quelles seraient les raies détectées avec cet algorithme.

Reprenons le cas de la figure 5 et voyons quels seraient les résultats dans ce cas.

- HNIV en A ⟶ pas de raie détectée.
- HNIV en B ⟶ une raie détectée en Y.
- HNIV en C ⟶ une raie détectée en (X+Y)/2.
- HNIV en D ⟶ une raie détectée en (X+Y)/2.

On peut trouver d'autres exemples de ce genre dans la partie centrale du spectre représenté page 25.

La page 30 représente l'organigramme de cet algorithme qui pourrait être facilement modifié pour s'adapter à des applications particulières.



NOTATIONS :

$\Delta\sigma$	pas d'échantillonnage du spectre et de la dérivée seconde BI. L'abcisse du premier échantillon est σ_1.
HNIV	niveau de détection.
NRAIE	numéro de la raie détectée en SIGS(NRAIE)

- Il y a en mémoire un bloc BI de 6144 valeurs de la dérivée seconde.

6110 → Débranchement en 6110 si on atteint la dernière valeur du tableau BI

ALGORITHME DE DETECTION DES RAIES

5) TEST DE PRECISION SUR LA POSITION DES RAIES

Plusieurs tests ont été faits à partir d'interférogrammes synthétiques correspondant à des spectres d'émission avec quelques raies bien isolées. Le calcul des valeurs de l'interférogramme a été fait en double précision (12 chiffres significatifs), mais le traitement complet (D.F.T., interpolation et pointé) a été fait avec la chaîne habituelle de nos programmes d'exploitation. Nous avons attribué aux valeurs du spectre ainsi calculé, des abcisses comparables à celles des spectres à haute résolution que nous traitons habituellement. Nous avons retenu deux exemples.

Exemple 1 : (Haute résolution mais faible étendue spectrale).

$M=20\ 000$ (nombre de points de l'interférogramme).
$h=0{,}45\ 10^{-2}$cm. (pas d'échantillonnage de l'interf. "DELTAD").
$5.\Delta\sigma=0.0034\ cm^{-1}$ (pas d'échantil. primaire du spectre "DELTAB").
$\sigma_T=16507\ cm^{-1}$ (position "théorique" de la raie).

$\sigma_{c1} = 16507.000003$ position de la raie mesurée par le programme "DERPTE".

Les écarts $\sigma_{ci} - \sigma_{c1}$ entre le milieu de la 1ère corde de la raie et le milieu de celle qui est le plus près du sommet sont :

$+ 3.10^{-6}$, $- 2.10^{-6}$, $- 5.10^{-6}$, $- 1.10^{-6}$, $- 3.10^{-6}$, $- 3.10^{-6}$, 3.10^{-6}, $- 2.10^{-6}$

Exemple 2 : (Haute résolution mais large étendue spectrale).

$M=2.10^6$
$h=0.4.10^{-4}$ cm
$5\Delta\sigma=0.0060\ cm^{-1}$
$\sigma_T=5000\ cm^{-1}$

$\sigma_{c1}=5000.000\ 00007$

Ecarts des autres cordes :

$- 7.10^{-8}$, $- 2.10^{-7}$, $- 3.510^{-7}$, $- 2.2.10^{-7}$, $+ 3.10^{-8}$, $+ 3.10^{-7}$, $+ 3.10^{-7}$

Ces deux exemples donnent une idée de la limite de précision qu'on ne peut dépasser du fait des incertitudes introduites par la chaîne de programmes qui, à partir de l'interférogramme, permet le calcul de la position des raies du spectre.

CETTE LIMITE SE SITUE ENTRE 10^{-9} et 10^{-10} EN VALEUR RELATIVE.

Avec ces mêmes programmes, et dans le but de faire une mesure de la vitesse de la lumière [38] G. GUELACHVILI a été amené a déterminer des nombres d'ondes absolus de raies à partir d'interférogrammes expérimentaux. Il s'agissait des raies d'absorption de la bande (2 ← 0) de CO. L'erreur quadratique moyenne sur la position de ces raies, calculée à partir de 7 spectres différents, est relativement égale à $2{,}5.10^{-8}$, ce qui correspond à une erreur absolu de l'ordre de 10^{-4} cm^{-1} pour des raies situées vers 4000 cm^{-1}. L'erreur systématique sur la position des raies ayant été évaluée à 10^{-8} en valeur relative, les nombres d'ondes absolus de ces raies ont pu être donnés avec une erreur relative de $3{,}5.10^{-8}$.

[38] GUELACHVILI (G.) Thèse - Faculté des Sciences d'Orsay - Mars 1973 -

CHAPITRE III

1. - Programme QUICK DERPTE

a) Cartes contrôle.

```
//QDERPTE JOB   ..............
/*SETUP UNIT=BD16,DDNAME=FT16F001,ID=(102946,NORING,SAVE,SL)
// EXEC FORTHLG,REGION.GO=115K
//LKED.SYSIN DD *
      .....
        .....
      ''DECK'' DU PROGRAMME QUICK-DERPTE.
          .....
            .....
/*
//GO.FT16F001 DD UNIT=BD16,DISP=OLD,LABEL=(1,SL),DSN=FT163037,
// DCB=(BUFNO=2,RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.SYSIN DD *
   11+0.3507984664000D-05    85900001
+0.197E+01+0.500E-03+0.250E+01
+0.62490000D+04+0.62800000D+04
/*
//
```

AUTRE EXEMPLE AVEC L'UTILISATION DE LA BIBLIOTHEQUE DES PROGR. SUR DISQUE.

```
//QDRP JOB  ................
//*          QUICK-DERPTE
//*        **************
//JOBLIB DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
/*SETUP UNIT=BD16,DDNAME=FT16F001,ID=(009530,RING,SAVE,BLP)
//GO EXEC PGM=QDRPTE,REGION=118K
//GO.FT16F001 DD UNIT=BD16,DISP=OLD,LABEL=(1,SL),DSN=FT16682,
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.FT05F001 DD DDNAME=SYSIN
//GO.FT06F001 DD SYSOUT=A
//GO.SYSIN DD *
   13+0.3507984664000D-05    99000001
+0.350E+01+0.100E-02+0.250E+01
+0.75800000D+04+0.76550000D+04
/*
//
```

b) Cartes données.

- Voir page suivante.

"QUICK-DERPTE" ⟹ {Détermination du "HNIV" optimum Limité à la recherche de 50 raies

Cartes

N°	1 5	7 10	15 20	25 30	35	40 45 50 55 60 65 70 72
1	IFR	FRANGE (D20.13)		M (I10)	LABEL	
2	GMAX (E10.3)	HNIV (E10.3)	FKGU(E10.3)			
3	SIG1 (D15.8)		SIG2 (D15.8)			

IFR = nombre de franges (FRANGE) par pas de l'interf (format I5).

FRANGE = valeur en CM^{-1} du pas élémentaire (le plus petit possible). Cette valeur peut être légèrement différente de celle utilisée lors du calcul (TFID-T101). Elle permet une redéfinition des abscisses du spectre. (ou plutôt une correction de ces dernières).

M = nombre de valeurs de l'interf. (voir la valeur prise au moment du calcul du TFID-T101 correspondant).

IABS = 1 pour un spectre d'absorption.

IABS = 0 pour un spectre d'émission ⟹ (G=3.0-G)

IAP = 1 apodisation du spectre $\left[(1-\frac{x^2}{x_M^2})^2\right]$ "APØ1" - en tenant compte de la valeur M donnée et de la puissance de 2 qui a servi au calcul de la T.F.
- si on donne à IAP les valeurs 2, 3 ou 4 l'apodisation sera de type "APØ2", "APØ3" ou "APØ4".

IAP = 0 pas d'apodisation.

GMAX = valeur de "GMAX" donnée par le programme "T101" (permet la normalisation des hauteurs de raies).

HNIV dès que la dérivée seconde du spectre dépasse le niveau HNIV, il y a détection d'une raie qui sera pointée (valeur moyenne : 0.01) (0.1 ≤ HNIV ≤ 0,0005 en gros).

SIG1, SIG2 limites en CM^{-1} du domaine spectral où on veut que se fasse la recherche automatique (prendre un domaine très réduit pour le QUICK DERPTE).

FKGU le domaine exploré $\Delta\sigma 2$ (voir fig. 35 p.5) est proportionnel à FKGU dont la valeur moyenne est 2.5
Si FKGU n'est pas compris entre 1 et 5 le programme impose la valeur 2.

REMARQUES :
- Il n'y a qu'une seule bande en Setup. Il s'agit de la bande (FT16) où se trouve le spectre calculé. Plus la valeur donnée à HNIV est petite, et plus il y a de raies détectées.
- Lorsque les raies sont déphasées, le repérage de la raie par la dérivée seconde est décalé systématiqueme Il faut alors prévoir une valeur de FKGU plus grande.

2. - Programme DERPTE

a) Cartes contrôle.

```
//DERPTE JOB   ..............
/*FORMAT PR,DDNAME=FT06F001,OVFL=ON,COPIES=1
/*SETUP DDNAME=FT14F001,UNIT=BD16,ID=104403
/*SETUP UNIT=BD16,DDNAME=FT16F001,ID=(102946,NORING,SAVE,SL)
//*ETUP UNIT=BD16,DDNAME=FT17F001,ID=(103827,RING,SAVE,SL)
// EXEC FORTHLG,REGION.GO=144K
//LKED.SYSIN DD *
     .....
       .....
     ''DECK'' DU PROGRAMME DERPTE.
         .....
           .....
/*
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2),
// LABEL=(5,SL)
//GO.FT16F001 DD UNIT=BD16,DISP=OLD,LABEL=(1,SL),DSN=FT163037,
// DCB=(BUFNO=2,RECFM=VS,LRECL=1028,BLKSIZE=1032)
//*O.FT17F001 DD UNIT=BD16,DISP=NEW,LABEL=(6,SL),DSN=FT173037,
//*DCB=(BUFNO=2,RECFM=VS,LRECL=404,BLKSIZE=408)
//GO.SYSIN DD *
   11+0.3507984664000D-05    859000010010
+0.100E+01+0.200E-02+0.250E+01
+0.76000000D+04+0.76300000D+04
+0.00000000D+00+0.00000000D+00 ((COEF CC1 ET CC2 (SIGC=SIG*(1+CC1)+CC2  ))
+0.30000000D+00+0.150E+0200005  ESSAI DU 15.02.72 SUR INTERF.3037
/*
//
```

AUTRE EXEMPLE AVEC L'UTILISATION DE LA BIBLIOTHEQUE DES PROGR. SUR DISQUE.

****

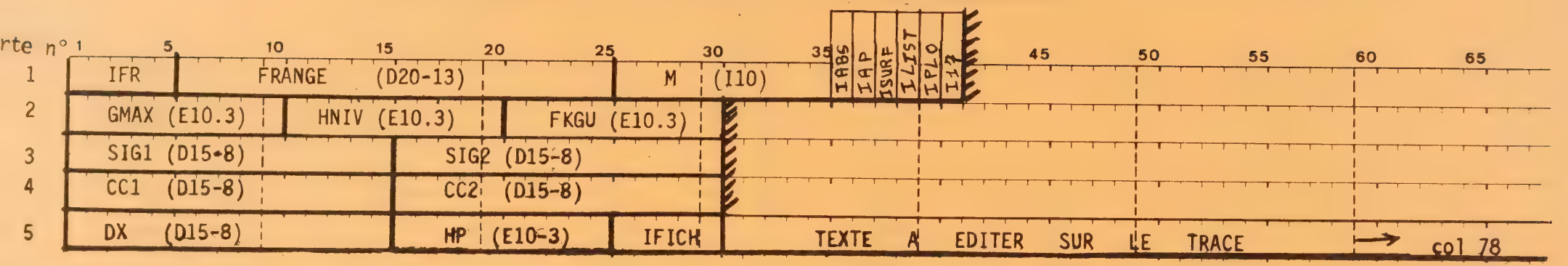
```
//DERPTE JOB   ..............
//*          DERPTE
//*         **********
/*MAIN LINES=40
/*FORMAT PR,DDNAME=FT06F001,OVFL=ON,COPIES=1
/*SETUP UNIT=BD16,DDNAME=FT16F001,ID=(009530,RING,SAVE,BLP)
//E EXEC FTHLG,PARM.LKED='NOLIST,NOXREF',REGION.GO=146K
//SYSLIB DD
//       DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE SYSLIB(DERPTE)
/*
//GO.FT16F001 DD UNIT=BD16,DISP=OLD,LABEL=(1,SL),DSN=FT16682,
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.SYSIN DD *
   13+0.3507984664000D-05    990000010000
+0.350E+01+0.500E-03+0.600E+01
+0.85000000D+04+0.11000000D+05
+0.00000000D+00+0.00000000D+00 ((COEF CC1 ET CC2 (SIGC=SIG*(1+CC1)+CC2  ))
+0.10000000D+00+0.280E+02    1
/*
//
```

REMARQUE : Les fichiers "FT14" et "FT17" sont facultatifs (voir cartes données). Dans le cas où ils ne sont pas utilisés, on peut soit supprimer les cartes contrôle correspondantes (/*SETUP et // DD) soit les faire précéder des 3 caractères //* en colonnes 1, 2, 3 ce qui permet alors au système de les ignorer.

b) Cartes données.

- Voir page suivante.

D E R P T E - (nombre illimité de raies)



- ILIST=1 Sortie détaillée
  ILIST=0 Sortie simplifiée
  } voir appendice 1 pour plus de détails

- IPLØ=0 Pas de tracé (supprimer ou faire commencer par //* les cartes "setup" et "DD" de la "FT14")

  IPLØ=1 Tracé du spectre et repérage des raies détectées

  IPLØ=2 Tracé du spectre et repérage des raies détectées + tracé de la dérivée seconde du spectre.

  - Pour IPLØ=1 ou 2, il faut rajouter la carte donnée n° 5 (pour IPLØ=0 elle est ignorée)

- I17=1 Résultats gardés sur la bande FT17.

  I17=0 Résultats non gardés (supprimer ou faire commencer par //* les cartes "Setup" et "DD" de la "FT17").

- ISURF=0 La surface et la largeur équivalente des raies pointées ne sont pas calculées.

  ISURF>0 La surface et la largeur équivalente des raies pointées sont calculées. Lors du pointé, chaque raie est étudiée sur un domaine de largeur $2.\Delta\sigma_2$ (voir introduction générale fig. 35). Si ISURF>0 (mais < 10), la largeur équivalente sera calculée et listée ISURF.fois. A chaque fois, la surface sera calculée sur un domaine spectral $2\Delta\sigma$ (voir fig. 37) qui sera égal à : $2* \left[\Delta\sigma_2 + k.\Delta\sigma_2/ISURF\right]$ avec k=0, 1, 2,....,ISURF-1. Le domaine $\Delta\sigma_3$ va donc en s'élargissant.

  $\Delta\sigma_2$ est calculé automatiquement en tenant compte de la largeur théorique de la fonction d'appareil. Toutes ces grandeurs sont d'ailleurs listées. *** Si ISURF>0, il est préférable de prendre ILIST=1.

CC1,CC2 Les abscisses des raies sont éventuellement corrigées par la formule : SIGC=SIG* (1 + CC1) + CC2

DX Echelle du tracé éventuel en CM^{-1}/CM de papier.

HP Hauteur totale du tracé en CM.

IFICH Numéro du fichier sur la bande "FT14"

} voir appendice 2, p. 22

* Pour les autres paramètres, se reporter aux cartes données du "QUICK-DERPTE"

3. - Exemples de durée du programme DERPTE

Pour l'encombrement en mémoire centrale des programmes QUICK DERPTE et DERPTE, voir le chapitre I,2.

a - QUICK - DERPTE

- 50 raies trouvées et pointées
- 7 blocs de 1024 valeurs interpolés

Temps ≃ 3 sec.
120 lignes
sur 370/165

b - DERPTE

- Interpolation de 360 blocs de 1024 valeurs primaires
- 17.000 raies trouvées
- ISURF=0 (HNIV=0.0005, PSMAX=0.05)
- Double liste (carte format)

Temps ≃ 6'27"
37.500 lignes
sur 360/75

c - DERPTE

- Interpolation de 715 blocs de 1024 valeurs primaires
- 19.512 raies trouvées
- ISURF=0, Double liste
- HNIV=0.0005, PSMAX=0.05

Temps ≃ 11'14"
39.500 lignes
sur 360/75

d - DERPTE

- Interpolation de 700 blocs de 1024 valeurs primaires
- 24.000 raies trouvées
- ISURF=0, ILIST=0, IPLØ=0
- Double liste

Temps ≃ 3'12"
48.300 lignes
sur 370/165

REMARQUE : Le 370/165 est environ 2,5 fois plus rapide que le 360/75.

CHAPITRE IV

Les formats de sortie du programme DERPTE

L'ensemble des programmes "DERPTE" commencent par sortir la liste des cartes données et un certain nombre de paramètres permettent de suivre la marche du programme.

Voici la signification des principaux :

NU → numéro de l'interprogramme

ND1024 → nombre de blocs de 1024 valeurs stockés sur "FT16"

NPRIM = ND1024 * 1024 = nombre de points du spectre avant l'interpolation.

SIG1B = abscisse en CM^{-1} du 1er point du spectre. Cette valeur apparaît deux fois car elle est recalculée au début en fonction de FRANGE.

DELTAB = pas d'échantillonnage du spectre avant interpolation.

BENGTH = 1024 * DELTAB en CM^{-1} = largeur d'un bloc.

DELTAG = DELTAB/5

NB1, NB2 = numéros des blocs (y compris) entre lesquels a été faite l'interpolation et la recherche.

a) QUICK-DERPTE ou ILIST=0 avec DERPTE

Il s'agit de la sortie la plus simple. Pour chaque raie trouvée, on trouvera :

1ère ligne :
- Numéro de la raie. (Qu'on retrouve sur le tracé éventuel).
- Abscisse en CM^{-1} du milieu de la corde la plus proche du sommet de la raie.
- Intensité = (Imax-Imin)* 10.000.
- Moyenne des écarts du centre des 8 cordes par rapport à celle qui est près du sommet.

b) DERPTE avec ILIST=1

Comme nous allons le voir la liste des résultats est dans ce cas plus complète et c'est celle qui doit être adoptée lorsque ISURF > 0. Pour chaque raie trouvée, on aura :

1ère ligne :
- Numéro de la raie. (Le même que sur le tracé éventuel).
- Abscisse de la raie (maximum de la dérivée seconde du spectre)
- Intensité minimum-Imin
- Intensité maximum-Imax
- Abscisse de la raie (milieu de la corde la plus proche du sommet) en CM^{-1}
- Abscisse précédente éventuellement corrigée par CC1 et CC2 (voir cartes données)
- Intensité = (Imax-Imin)/GMAX

2ème ligne : - 8 valeurs qui correspondent aux écarts des milieux des cordes par rapport à celle du sommet.

C'est la valeur de cette dernière qui est retranchée des autres (en CM^{-1})

3ème ligne : (si ISURF > 0)

- Domaine d'étude de la raie = largeur totale en CM^{-1} sur laquelle on mesure la surface de la raie.

- "Background" = niveau moyen du spectre estimé dans le domaine d'étude de la raie (=valeur moyenne du plus grand échantillon sur le flan droit, puis sur le flan gauche de la raie)

- largeur équivalente en CM^{-1}

Les lignes suivantes sont la répétition de cette 3ème ligne pour des domaines d'étude supérieurs (voir dans cartes données les explications fournies à propos de ISURF).

Sur les deux pages suivantes, on peut voir deux exemples de liste suivant que le paramètre ILIST vaut 0 ou 1.

c) LA CARTE FORMAT

Il s'agit d'une carte contrôle du sous système IBM ASP.

/*FORMAT PR,DDNAME=FT06F001,OVFL=ON,COPIES=n

- Le paramètre "OVFL=ON" permet un saut automatique de la liste des résultats à chaque changement de page ce qui évite d'avoir à lire sur la pliure du papier.

- n est le nombre de listes identiques désirées. Permet de distribuer la liste des résultats à différents utilisateurs. Pour une liste unique normale prendre n=1.

DERPTE SI ILIST=0 OU QUICK DERPTE

1.	6351.680684	3119.	C.0069
2.	6352.192513	2717.	C.0027
3.	6353.119246	2135.	C.0063
4.	6354.517860	2743.	-0.0192
5.	6355.934629	3479.	C.0151
6.	6356.329328	3167.	C.0053
7.	6357.323686	3266.	C.0079
8.	6357.851392	4752.	C.0070
9.	6358.687713	2227.	C.0075
10.	6359.104454	2345.	-C.0396
11.	6359.975682	2949.	C.0020
12.	6360.683789	4255.	C.0008
13.	6361.254195	2533.	C.0117
14.	6361.698678	4113.	-C.0014
15.	6362.399933	5608.	C.0174
16.	6363.734974	2779.	C.0122
17.	6364.933954	2537.	C.0201
18.	6365.982037	2784.	C.0242
19.	6367.202642	2306.	-C.0207
20.	6368.052452	1834.	C.0220
21.	6368.356892	1850.	-C.0421
22.	6369.088611	3872.	C.0096
23.	6369.177676	2909.	C.1480
24.	6371.952422	1429.	-0.0752
25.	6375.822573	1331.	C.0037
26.	6376.188763	1865.	-C.0513
27.	6376.806558	2974.	-C.0015
28.	6377.827008	1677.	C.0153
29.	6378.886753	2081.	C.0045
30.	6381.305268	1680.	C.0089
31.	6381.952292	1484.	-C.0047
32.	6382.861025	1885.	-C.0058
33.	6384.118923	3219.	C.0032
34.	6384.652872	1285.	-C.0182
35.	6385.347374	1792.	-0.0038
36.	6386.741926	1989.	C.0031
37.	6387.569345	3073.	C.0023
38.	6388.237995	3745.	-C.0128
39.	6388.750562	2390.	0.0063
40.	6389.331053	2183.	C.0158
41.	6389.613862	1800.	-0.0493
42.	6390.021246	1744.	-0.0151
43.	6391.758658	2087.	-C.0211
44.	6394.140207	2608.	0.0140
45.	6396.386852	2321.	-0.0134
46.	6397.181488	1630.	-0.0158
47.	6398.341269	4647.	-C.0499
48.	6398.417563	4687.	C.0675
49.	6399.294887	2111.	C.0072
50.	6400.606824	2797.	-C.0055
51.	6401.257604	2509.	-0.0113
52.	6402.498433	2942.	-C.0131
53.	6403.227920	2075.	-C.0135
54.	6404.359169	2170.	C.0046
55.	6404.975854	2541.	-C.0014
56.	6406.073397	2484.	-C.0129
57.	6406.715418	2345.	C.0059
58.	6407.719240	2657.	C.0002
59.	6408.310085	3259.	-C.0035

6351.6709 1.5675 2.4096 6351.680684 6351.680684 0.3119
0.00125 0.00262 0.00420 0.00619 0.00881 0.01272 0.01519 0.00339
44761717E 00 0.22934046E 01 0.76348782E-01
59682286E 00 0.23516407E 01 0.86161673E-01
74602854E 00 0.23516407E 01 0.89662313E-01

6352.1850 1.5365 2.2702 6352.192513 6352.192513 0.2717
0.00031 0.00064 0.00113 0.00172 0.00248 0.00349 0.00488 0.00711
44761717E 00 0.22307634E 01 0.72873592E-01
59682286E 00 0.22842350E 01 0.82882047E-01
74602854E 00 0.23196812E 01 0.96275210E-01

6353.1236 2.0390 2.6155 6353.119246 6353.119246 0.2135
0.00011 0.00031 0.00072 0.00140 0.00259 0.00543 0.02579 0.01372
44761717E 00 0.25490885E 01 0.43950606E-01
59682286E 00 0.25766096E 01 0.49055409E-01
74602854E 00 0.25794935E 01 0.53647138E-01

6354.5427 1.7993 2.5400 6354.517860 6354.517860 0.2743
-0.00421 -0.01098 -0.02520 -0.04089 -0.03268 -0.02394 -0.01411 -0.00167
44761717E 00 0.23134918E 01 0.51896643E-01
59682286E 00 0.23797045E 01 0.64324915E-01
74602854E 00 0.24528551E 01 0.81873953E-01

6355.9283 1.6307 2.5701 6355.934629 6355.934629 0.3479
0.00198 0.00535 0.01337 0.03890 0.03013 0.02091 0.01090 -0.00056
44761717E 00 0.22686815E 01 0.62568307E-01
59682286E 00 0.23523302E 01 0.82244694E-01
74602854E 00 0.23792543E 01 0.10636789E 00

6356.3305 1.1837 2.0387 6356.339328 6356.339328 0.3167
0.00054 0.00117 0.00203 0.00310 0.00448 0.00637 0.00933 0.01540
44761717E 00 0.19947996E 01 0.92472732E-01
59682286E 00 0.20302963E 01 0.10057569E 00
74602854E 00 0.20721827E 01 0.12108028E 00

6357.3138 1.8280 2.7098 6357.323686 6357.323686 0.3266
0.00069 0.00161 0.00287 0.00469 0.00774 0.01480 0.02168 0.00923
44761717E 00 0.25800428E 01 0.63690245E-01
59682286E 00 0.25972834E 01 0.67106843E-01
74602854E 00 0.25972834E 01 0.76854110E-01

6357.8278 1.0441 2.3270 6357.851392 6357.851392 0.4752
0.00214 0.00458 0.00740 0.01070 0.01456 0.01564 0.00595 -0.00467
44761717E 00 0.20591469E 01 0.12533741E 00

DERPTE SI ILIST=1 et ISURF=3

2. - Tracé éventuel du spectre et de sa dérivée seconde

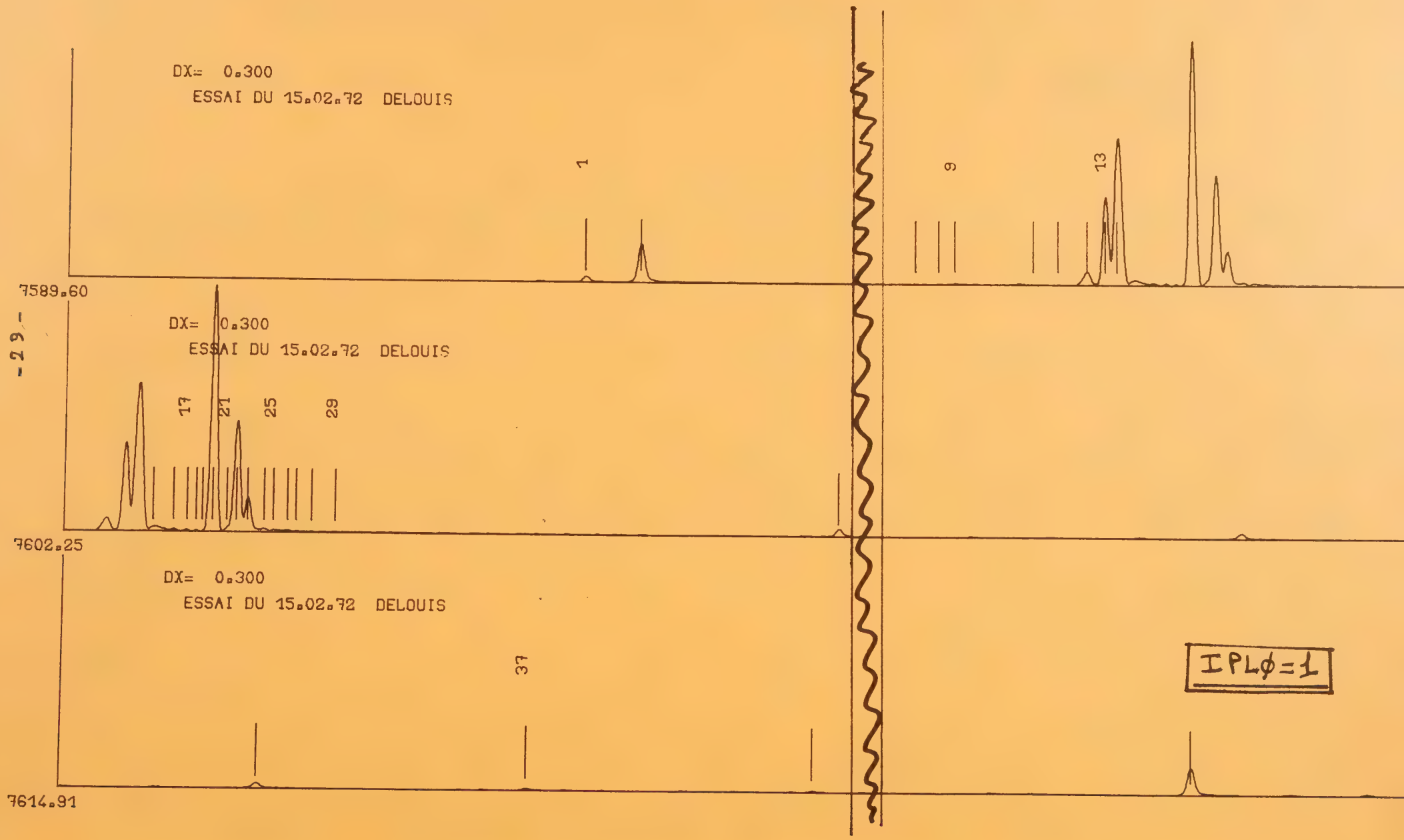
Nous avons vu à propos des cartes données que si le paramètre "IPLØ" avait la valeur 1 le spectre était tracé et que les raies détectées y étaient repérées par un trait vertical numéroté, le numéro étant le même que celui indiqué sur la liste des résultats. Les pages 29 et 30 suivantes nous montrent comment sont faits ces tracés. Chaque bloc "GG" de 6144 valeurs du spectre interpolé (voir figure 38, page 11) est tracé d'un seul bloc et trois blocs consécutifs sont tracés les uns au dessous des autres en partant du haut sur une page avant de passer à la page suivante. L'utilisateur peut fixer l'échelle en abscisses (tracée en face de la valeur DX) et la hauteur totale de la page. Comme on le voit sur la page 29, l'utilisateur peut aussi faire tracer un titre en tête de chaque bloc et il existe une zone de recouvrement entre chaque bloc pour assurer la liaison entre ces derniers. L'abscisse du début de chaque bloc est indiqué en CM^{-1}.

Si "IPLØ" a la valeur 2, le programme trace aussi la dérivée seconde et le niveau "HNIV" de détection. Le tracé de la dérivée seconde renormalisé dans chaque bloc ce qui explique les différences de niveau apparent pour "HNIV" sur la page 31. La dérivée seconde est tronquée vers le bas. Le tracé de la dérivée seconde peut se révèler utile dans le cas de l'étude de raies aux formes complexes et dissymétriques, comme c'est le cas au centre de la page 31. (voir paragraphe II.4 de cette brochure).

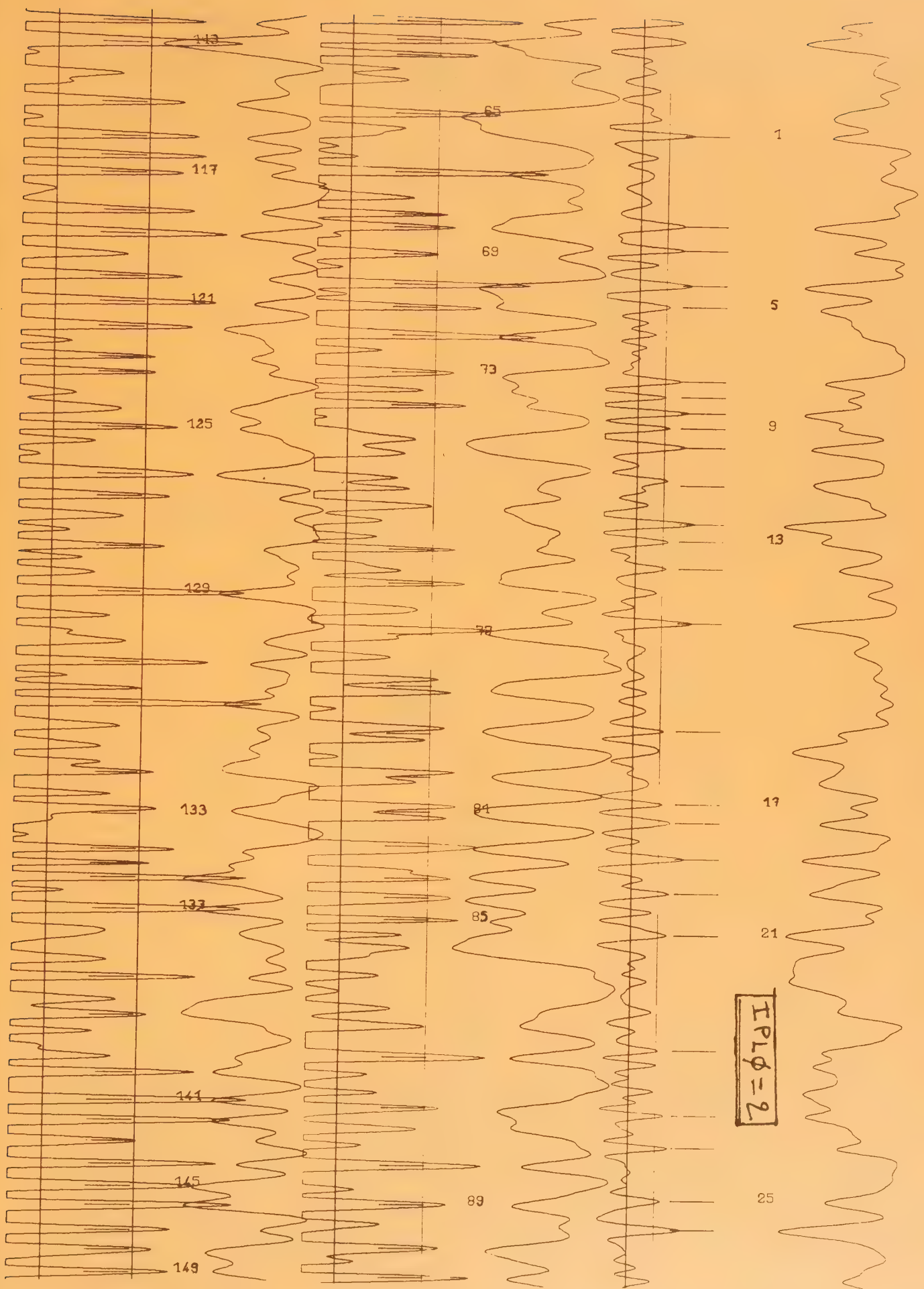
S'il y a n blocs à tracer, le tracé de la dérivée seconde commencera aux"numéros d'études"2n + 1 ce qui peut permettre un changement de plume pour la différencier du spectre proprement dit.

Dans le cas où il n'y a qu'un seul bloc à tracer, il faudra demander le tracé à partir du bloc 2 pour éviter le positionnement dans le haut de la page.

"Numéros d'études" : il s'agit de numéros écrits par le programme sur la bande magnétique "FT14" à tracer. Ces numéros sont reconnus par le lecteur du traceur et ils peuvent donc servir au positionnement de la bande au moment du tracé.


DX= 0.300
ESSAI DU 15.02.72 DELOUIS
1
9
13
7589.60
DX= 0.300
ESSAI DU 15.02.72 DELOUIS
17
21
25
29
7602.25
DX= 0.300
ESSAI DU 15.02.72 DELOUIS
37
IPLφ=1
7614.91

6352.85
DX= 1.000
281
285
I=1 PL=1
6395.67
DX= 1.000
233
237
241
245
249
253
257
281
6338.42
DX= 1.000
177
181
185
189
193
197
201


113
117
121
125
129
133
137
141
145
149
65
69
73
77
81
85
89
1
5
9
13
17
21
25
IPL φ=2

3. - Sauvegarde des résultats sur la bande magnétique "FT17"

C'est l'équivalent de la liste simplifiée (ILIST=0) qui est sauvegardée. Ces résultats sont regroupés par blocs de 100 raies.

Pour relire la bande "FT17" il suffirait de faire le programme suivant :

```
      REAL*8  SIM(4,100)
      DØ 99 NBLØC=1,999
   99 READ (17,END=888)SIM
  888 STØP
      END
```

avec SIM(1,J)=numéro de la raie J.
SIM(2,J)=abscisse de la raie en CM^{-1}
SIM(3,J)=intensité de la raie normalisée puis multipliée par 10.000.
SIM(4,J)=moyenne des écarts de dissymétrie de la raie pointée.

- Si on ne désire pas garder ces résultats, il suffira d'enlever les cartes contrôles relatives à la bande "FT17" et de donner la valeur zéro au paramètre "I17" des cartes données.

4. - Remarque à propos des intensités des raies

Les valeurs du spectre sur lesquelles le programme DERPTE travaille sont celles du spectre primaire (sorties de la T.F. sans apodisation et non interpolées) auxquelles on a fait subir trois transformations successives :

a) Normalisation à l'unité par division de toutes les valeurs par la plus grande positive.

b) Interpolation et apodisation qui a pour effet secondaire de multiplier toutes les valeurs normalisées précédentes par un coefficient constant "GMAX" qui est listé par le programme "T101", et dont la valeur est comprise entre 1 et 3 selon les cas.((*))

c) Division de toutes les valeurs interpolées par la valeur de "GMAX" qu'il trouve dans les cartes données.

REMARQUE : Pour les spectres d'émission, les valeurs interpolées "G" non encore divisées par "GMAX" sont changées en "G"=(3.0-G) pour se ramener au cas de l'absorption.

((*)) "T101" et "T102" : programme de calcul des valeurs pré-primaires du spectre (voir brochure "le manuel du parfait petit fouriériste").

```
C  ******PROGRAMME ''DERPTE '' (NOMBRE DE RAIES ILLIMITE.)  ********************
C
      REAL*8 PET1,PET2,HDA,HDB,HDAB,SIGCC,RNNGS2,SIGD,RNND,DSIGS2,RM    SSSSSSS1
      REAL*8 SIGC01,SIGS,SIGG5,SIGG6,SIGMA,HAUT,GGMIN
      REAL*8 GGMAX,DGMAX,SIGGS,SIM,RNB2L,CC1,CC2
      REAL*8 SIG1B,DELTAB,DELTAG,BENGTH,RNFM,SIGLIM,DELTAD,FRANGE
      REAL*8 RIFR,TRANCH,RMR,RNSB,SIG1G
      REAL*8 SIGA,SIGB,SIGIG,GARDE,FJ,SIGIG1,FKP
      DIMENSION B(1050),AA(109),H(5120),BB(1024),SIM(400),G(200)
      DIMENSION HAUT(10),SIGS(10),SIGC01(10),DSIGD1(10),DSIGZ3(10)
      DIMENSION GG(6144),SIGMA(6144),BI(6144),SIGGS(250)
      DIMENSION IBUF(1013),TEXT(12)
      EQUIVALENCE(SIGMA(1),BI(1)),(H(1),GG(1025)),(B(21),BB(1))
      DATA MNIV/9/,NSIMAX/250/,PET1/0.1D-10/,PET2/0.1D-12/,DSIGZ3/10*0./
      DATA LR/0/,KJI/1/,IR/1/,PSMAX/0.424E-01/
C
C     NSIMAX=NBRE MAX. DE RAIES PAR BLOC =DIMENSION EN REAL*8 DE SIGGS.
      PRINT 1900
 1900 FORMAT(' ***** PROGR. DERPTE V.22/08/72 *****',/)
      READ (16)NU,NU,NU,ND1024,SIG1B,DELTAB
      NPRIM=ND1024*1024
      RNSB=NPRIM
      SIGB=SIG1B+512.*DELTAB
      PRINT 1024,NU,ND1024,NPRIM,SIG1B,DELTAB
 1024 FORMAT('  NU=',I8,' ND1024=',I10,' NPRIM=',I10,' SIG1B=',D15.8,' D
     1ELTAB=',D15.8,/)
C
C    CALCUL DE NOUVELLES VALEURS DE SIG1B ET DE DELTAB EN FONCTION DE LA VALEUR
C     FRANGE LUE SUR  LES CARTES DONNEES
      READ(5,197)IFR,FRANGE,M,IABS,IAP,ISURF,ILIST,IPLO,I17
  197 FORMAT(I5,D20.13,I10,6I1)
      READ(5,6060)GMAX,MNIV,FKGU                                        SSSSSSS1
 6060 FORMAT(3E10.3)                                                    SSSSSSS1
      IF(FKGU.LT.0.9)FKGU=2.                                            SSSSSSS1
      IF(FKGU.GT.6.1)FKGU=2.
      PRINT 115,FKGU                                                    SSSSSSS1
  115 FORMAT(' FKGU=',F6.2,'  VAL. MOY.=2.')                            SSSSSSS1
      PRINT 28,FRANGE,IFR,SIGB,IAP,M,IABS,GMAX,MNIV,ISURF,ILIST,IPLO,I17
   28 FORMAT(' FRNGE',D20.13,' IFR=',I5,' SIGB=',D15.8,' IAP=',I1,' M=',
     1I8,' IABS=',I1,/,'  GMAX=',E10.3,' MNIV=',E10.3,' ISURF=',I1,' ILI
     1ST=',I1,' IPLO=',I1,' I17=',I1,/)
      RNDM=GMAX/10000.
      RIFR=IFR
      DELTAD=FRANGE*RIFR
      WRITE(6,8877)DELTAD
 8877 FORMAT(/,' LE PAS D ECHANTILLONNAGE DE L INTERF. SOIT DELTAD,EST
     1EGAL A ',D15.8,'CM',/)
      TRANCH=1./(2.*DELTAD)
      UNS2M=RNSB*DELTAB
      NFM=(TRANCH/UNS2M)+0.5
      IF(NFM.EQ.1)GO TO 128
      RNFM=NFM
      DELTAD=DELTAD*RNFM
      TRANCH=TRANCH/RNFM
  128 MR=SIGB*DELTAD
      RMR=MR
      DSIGB=SIGB-(RMR/DELTAD)
      IF(DSIGB.GT.TRANCH)IR=3
      DELTAB=1./(2.*DELTAD*RNSB)
      DELTAG=DELTAB/5.
      BENGTH=5120.*DELTAG
      SIG1B=RMR/DELTAD
      IF(IR.NE.1)SIG1B=SIG1B+TRANCH
      PRINT 129,NFM,MR,IR,SIG1B,DELTAB,BENGTH,DELTAG
  129 FORMAT(/,' NFM=',I8,' MR=',I4,' IR=',I4,' SIG1B=',D15.8,' DELTAB='
     1,D15.8,/,' BENGTH=',D15.8,' DELTAG=',D15.8,/)
C
C *** CALCUL DE LA FONCTION D INTERPOLATION AA
C *** AA=TF. D UN RECT. DE LONG. 1 ECHANT. AVEC  201 VAL.(DELTAD=0.005)
C    L'ARG. DU COS DANS LA TF. EST EGAL A 2*PI*SIG*DELTAD AVEC DELTAD=FIK=0.005
C    ET SIG=FIJ*(1/2*L/5)=FIJ*0.1        (SIA=2*PI*DELTAD)
C *** CE RECT. FEUT ETRE APODISE SUR MP=( 201*M)/NPRIM VALEURS SI IAP=1
      MP=201
      DO 1 I=2,200
    1 B(I)=1.0
      B(201)=0.5
C  *** APODISATION DE L INTERF
      IF(IAP.EQ.0)GO TO 712
      MP=(201*M)/NPRIM
      Q=(MP-1)*(MP-1)
      IF(IAP-2)1935,1936,1937                                           22/08/72
 1935 R=0.                                                              APO1   *
      S=0.
      T=1.
      W=0.
      V=2.
      GO TO 5000
 1936 R=0.06267                                                         APO2
      S=0.33213
      T=0.21839
      W=0.38680
      V=2.
      GO TO 5000
 1937 IF(IAP-3)1939,1939,1940
 1939 R=0.23978                                                         APO3
      S=0.45808
      T=0.22439
      W=0.077208
      V=2.
      GO TO 5000
 1940 R=0.                                                              APO4
      S=0.
      T=1.
      W=0.
      V=5./2.
 5000 DO 39 I=2,MP
      FI=I-1
      F=1.-(FI*FI)/Q
      PK=R+S*F+T*F**V+W*F**3                                                *
   39 B(I)=B(I)*PK                                                      22/08/72
      PRINT 1900
C *** CALCUL TF CLASSIQUE  AA=TF.(B)
  712 AMAX=0.5
      DO 4 IK=2,MP
    4 AMAX=AMAX+B(IK)
      SIA=+0.3141592653D-02
      DO 9 IJ=56,105
      FIJ=IJ-55
      SIG=SIA*FIJ
      AA(IJ)=0.5
      DO 3 IK=2,MP
      FIK=IK-1
    3 AA(IJ)=AA(IJ)+B(IK)*COS(SIG*FIK)
C *** NORMALISATION ET APODISATION DE LA FONCTION AA
    9 AA(IJ)=(AA(IJ)/AMAX)*(1.-(FIJ*FIJ)/2500.)**2
      AA(55)=1.
      DO 908 I=106,109
  908 AA(I)=0.
C *** DEFINITION PAR SYMETRIE DE LA FCT. D'INTERP. AA COMPLETE
      DO 181 I=1,54
  181 AA(I)=AA(110-I)
      PRINT 183,MP
  183 FORMAT(/,' FCT. D INTERPOLATION AA AVEC MP=',I5)
C
C ***CARACTERISTIQUES DE L'ETUDE QUI DOIT ETRE FAIT DU BLOC NB1 OU SE TROUVE
C *** SIG1 A CELUI OU SE TROUVE SIG2.
      READ(5,90)SIG1,SIG2
   90 FORMAT(4D15.8)
      PRINT 185,SIG1,SIG2
  185 FORMAT(2X,'SIG1=',D15.8,' SIG2=',D15.8)
      GARDE=40.*DELTAB
C     SIG1=SIG1-GARDE
C     SIG2=SIG2+GARDE
      READ(5,90)CC1,CC2
      PRINT 308,CC1,CC2
  308 FORMAT(/,' CORRECT. DES ABCISSES PAR LES COEF CC1=',D15.8,' CC2=',
     1D15.8,' SIGC=SIG*(1.+CC1)+CC2',/)
      NB1=(SIG1-SIG1B)/BENGTH+1.00001
```

```
      NB2=(SIG2-SIG1B)/BENGTH+1.00001
      IF(NB1.LE.0)NB1=1
      IF(NB2.LE.0)NB2=1
      IF(NB2.GT.ND1024)NB2=ND1024
      IF(NB1.GT.NB2)NB1=NB2
      IF(NB1.EQ.1) GO TO 187
      NB1M1=NB1-1
      DO 188 I=1,NB1M1
  188 READ(16)BB
      DO 189 I=1025,1044
      J=I-1024
  189 B(J)=B(I)
      GO TO 190
  187 DO 191 I=1,20
  191 B(I)=0.0
  190 RNB1M1=NB1-1
      NB=NB2-NB1+1
      RNB2L=NB
      SIG1G=SIG1B+RNB1M1*BENGTH-10.*DELTAB
      SIGA=SIG1G+40.*DELTAB
      SIGIG=SIG1G-1024.*DELTAG
      SIGIG1=SIGIG
      SIGB=SIGIG+BENGTH+GARDE
      SIGLIM=SIG1G+RNB2L*BENGTH-GARDE
C   ON DOIT INTERPOLER DU BLOC NB1 AU BLOC NB2
      PRINT 208,SIG1G,NB1,NB2,SIGLIM,PSMAX
  208 FORMAT(/,' SIG1G=',D15.8,' NB1=',I8,' NB2=',I8,/,' SIGLIM=',F12.4,
     1' PSMAX=',E10.3,/)
C
C   CHAQUE RAIE POUR SON POINTAGE EST ETUDIEE SUR UN DOMAINE DE 2*DSIGS2 CM-1
C   LA LARGEUR A MI HAUTEUR D'UNE RAIE SYNTHETIQUE EST EGALE A 2*DELTAB CM-1
C   SI M=NPRIM,D'OU LA VALEUR PRISE POUR DSIGS2.
      RM=M
      DSIGS2=DELTAB*(RNSB/RM)*FKGU                                    SSSSSSS1
      PRINT 47,DSIGS2,IABS
   47 FORMAT('         RAIES A POINTER SUR UN INTERVALLE DE + OU -',D12.5,
     1,' CM-1                IABS=',I2)
C
C   SI ISURF >0 LA LARGEUR EQUIVALENTE DE CHAQUE RAIE SERA CALCULEE SUR UN
C   DOMAINE 2*DSIGZ3(I) OU I PREND LES VALEURS 1,2,3,...ISURF ET DSIGZ3(I) LES
C   VALEURS DSIGS2+(I-1)*(DSIGS2/ISURF) AVEC ISURF INF. OU EGAL A 10
      IF(ISURF.LT.1)GO TO 2
      IF(ISURF.GT.10)ISURF=10
      FSURF=ISURF
      DDS=DSIGS2/FSURF
      DO 5 I=1,ISURF
      FI=I-1
    5 DSIGZ3(I)=DSIGS2+FI*DDS
      PRINT 30,DSIGZ3
   30 FORMAT(/,15X,'DSIGZ3',/,1X,10E12.4)
    2 NNGS2=DSIGS2/DELTAG
      RNNGS2=NNGS2
C
C     INITIALISATION DU TRACE
      IF(IPLO.EQ.0)GO TO 6
      READ(5,11)DX,HP,IFICH,TEXT
   11 FORMAT(D15.8,E10.3,I5,12A4)
      PRINT 12,DX,HP,IFICH,TEXT
   12 FORMAT(' DX=',D15.8,' HP=',E10.3,' IFICH=',I5,12A4)
      CALL IBENA(IBUF,1013,14)
      IP=3
      DSIGMA=6143.*DELTAG/DX
      HT=HP/3.33
      HMAX=HT+0.75
      HMIN=-0.75
      HREF=HT/2.
      H1=HREF-1.
      H2=HREF-2.
      NREF=4
      HN=HT/GMAX
      DELTAX=DELTAG/DX
      NETUD=1
      DDX=DX
      X1=DSIGMA+2.
      CALL PNUMA(0.,0.,NETUD,X1,-1.)
C
C
C
    6 DO 192 KP=1,NB
C
C   INITIALISATION DU TRACE POUR CHAQUE BLOC INTERPOLE.
      IF(IPLO.EQ.0)GO TO 8
      IP=IP+1
      IF(IP.LT.4)GO TO 9
      IP=1
      Y1=HP*2./3.
      X1=DSIGMA+2.
      GO TO 10
    9 Y1=-HP/3.
      X1=0.
   10 CALL PNUMA(X1,Y1,NETUD,0.,0.)
      CALL PCARA(2.,HT-1.,0,TEXT,48,0.2,0.2,1.,0.)
      CALL PCARA(2.,HT-0.5,0,'DX=',3,0.2,0.2,1.,0.)
      CALL NOMBA(3.,HT-0.5,0,DDX,3,0.2,0.2,1.,0.)
      CALL TRAA(0.,HT,0)
      CALL TRAA(0.,0.,1)
      SG=SIGIG
      CALL NOMBA(-1.,-0.4,0,SG,2,0.2,0.2,1.,0.)
      CALL TRAA(0.,0.,0)
C
C
C
    8 READ(16)BB
C
C     CONVOLUTION
C
      DO 194 K=1,5120
      I=(K-1)/5+1
      J=I*5-K+1
  194 H(K)=B(I)*AA(J)+B(I+20)*AA(J+100)+B(I+1)*AA(J+5)+B(I+19)*AA(J+95)+
     1B(I+2)*AA(J+10)+B(I+18)*AA(J+90)+B(I+3)*AA(J+15)+B(I+17)*AA(J+85)+
     2B(I+4)*AA(J+20)+B(I+16)*AA(J+80)+B(I+5)*AA(J+25)+B(I+15)*AA(J+75)+
     3B(I+6)*AA(J+30)+B(I+14)*AA(J+70)+B(I+7)*AA(J+35)+B(I+13)*AA(J+65)+
     4B(I+8)*AA(J+40)+B(I+12)*AA(J+60)+B(I+9)*AA(J+45)+B(I+11)*AA(J+55)+
     5B(I+10)*AA(J+50)
      DO 196 I=1,20
      J=I+1024
  196 B(I)=B(J)
      IF(IABS.EQ.1)GO TO 4203
C   CAS DES SPECTRES D'EMISSION.
      DO 23 I=1,5120,2
      H(I)=3.-H(I)
   23 H(I+1)=3.-H(I+1)
 4203 NRAIE=0
C
      IF(KP.NE.1)GO TO 29
      FI=GG(1025)
      DO 32 I=1,1024
   32 GG(I)=FI
C
C     TRACE D'UN BLOC DE 6144 VALEURS.
   29 IF(IPLO.EQ.0)GO TO 13
      IF(IABS.EQ.1)GO TO 14
      DO 17 I=1,6144
      FI=I-1
      X=FI*DELTAX
      Y=(3.-GG(I))*HN
      IF(Y.GT.HMAX)Y=HMAX
      IF(Y.LT.HMIN)Y=HMIN
   17 CALL TRAA(X,Y,1)
      GO TO 18
   14 DO 15 I=1,6144
      FI=I-1
      X=FI*DELTAX
      Y=GG(I)*HN
      IF(Y.GT.HMAX)Y=HMAX
      IF(Y.LT.HMIN)Y=HMIN
   15 CALL TRAA(X,Y,1)
```

```
   18 CALL TRAA(X,0.,0)
      CALL TRAA(0.,0.,1)
C
C
C    B    CALCUL DE LA DERIVEE SECONDE NORMALISEE EN BI.
C    RECHERCHE DE L'ABCISSE EN CM-1 DES RAIES  DONT LA DERIVEE SECONDE DEPASSE
C    LE NIVEAU ''HNIV'' ET STOCKAGE DES '' NRAIE '' RAIES AINSI TROUVEES DANS
C    LE TABLEAU SIGGS.
   13 BI(1)=0.
      BI(2)=0.
C
C    TRACE DE LA DERIVEE SECONDE ET DE HNIV.
      IF(IPLO.NE.2)GO TO 24
      BIMX=0.
      DO 6007 I=3,6144
      BI(I)=(GG(I-2)-2.*GG(I-1)+GG(I))/PSMAX
 6007 IF(BI(I).GT.BIMX)BIMX=BI(I)
      HM=HT/BIMX
      CALL PNUMA(0.,0.,NETUD,0.,0.)
      DO 20 I=3,6144
      RI=I-2
      X=RI*DELTAX
      Y=BI(I)*HM
      IF(Y.LT.HMIN)Y=HMIN
   20 CALL TRAA(X,Y,1)
      HDETEC=HNIV*HM
      CALL TRAA(X,HDETEC,0)
      CALL TRAA(0.,HDETEC,1)
      CALL TRAA(0.,0.,0)
C
      GO TO 25
   24 DO 26 I=3,6144
   26 BI(I)=(GG(I-2)-2.*GG(I-1)+GG(I))/PSMAX
C
   25 SIGC2=0.
      I=0
   36 I=I+1
      IF(I.EQ.6144)GO TO 6110
      IF(BI(I).LE.HNIV)GO TO 36
   38 I=I+1
      IF(I.EQ.6144)GO TO 6110
      IF(BI(I).GT.BI(I-1))GO TO 38
      FI=I-3
      SIGCC=SIGIG+FI*DELTAG
   41 IF(BI(I).LT.0.)GO TO 37
      I=I+1
      IF(I.EQ.6144)GO TO 6110
      IF(BI(I).LT.BI(I-1))GO TO 41
   40 I=I+1
      IF(I.EQ.6144)GO TO 6110
      IF(BI(I).GE.BI(I-1))GO TO 40
      FI=I-3
      SIGC2=SIGIG+FI*DELTAG
      GO TO 41
   37 IF(SIGC2.LT.0.1)GO TO 42
      SIGCC=(SIGCC+SIGC2)/2.
      SIGC2=0.
   42 IF(SIGCC.LT.SIGA)GO TO 36
      IF(SIGCC.GT.SIGB)GO TO 6110
      NRAIE=NRAIE+1
      SIGGS(NRAIE)=SIGCC
C
C     REPERAGE DES RAIES POINTEES
      IF(IPLO.EQ.0)GO TO 19
      NREF=NREF+1
      X=(SIGCC-SIGIG)/DX
      CALL TRAA(X,H2,0)
      CALL TRAA(X,H1,1)
      IF(NREF.LT.5)GO TO 19
      NREF=1
      FREF=LR+NRAIE
      CALL NOMBA(X,NREF,0,FREF,0,0.2,0.2,0.,1.)
C
   19 IF(NRAIE.EQ.NSIMAX)GO TO 6111
      GO TO 36
C
C
C
C=====================================================================
C
C    PARTIE DU PROGRAMME QUI POINTE LES ''NRAIE'' RAIES ENTRE SIGA ET SIGB.
C
 6110 DO 31 I=1,6144
      FJ=I-1
   31 SIGMA(I)=SIGIG+FJ*DELTAG
      IF(NRAIE.EQ.0)GO TO 185
C
C
      DO 67 LJ=1,NRAIE
C
C     CALCUL DE GGMIN ET DE GGMAX ET DES NIVX.
C
      LR=LR+1
C  *** RECHERCHE POUR UNE DES RAIES DE NNP ET NND
      SIGD=SIGGS(LJ)+RNNGS2*DELTAG
      RNND=((SIGD-SIGIG)/DELTAG)+1.5
      NND=RNND
      NNP=NND-(2*NNGS2)
      GGMIN=GG(NNP)
C
C  *** RECHERCHE POUR UNE DES RAIES DE GGMAX ET GGMIN
      JJM=NNP
      DO 68 JJ=NNP,NND
      IF(GGMIN-GG(JJ))68,68,78
   78 GGMIN=GG(JJ)
      JJM=JJ
   68 CONTINUE
      GGMAXG=GG(NNP)
      JJMAXG=NNP                                                    1/7/72
      GGMAX=GG(NND)
      DO 79 JJ=NNP,JJM
      IF(GG(JJ).LT.GGMAXG)GO TO 79                                  1/7/72
      GGMAXG=GG(JJ)                                                 1/7/72
      JJMAXG=JJ                                                     1/7/72
   79 CONTINUE                                                      1/7/72
      NNP=JJMAXG                                                    1/7/72
C  79 IF(GG(JJ).GT.GGMAXG)GGMAXG=GG(JJ)
      DO 81 JJ=JJM,NND
   81 IF(GG(JJ).GT.GGMAX)GGMAX=GG(JJ)
      IF(GGMAXG.LT.GGMAX)GGMAX=GGMAXG
      DGMAX=GGMAX-GGMIN
      IF(DGMAX.LT.0.0001)DGMAX=0.0001
      SIM(KJI+2)=DGMAX/RNDM
      SIM(KJI+3)=0.
      DO 82 JJ=1,9
      RJJ=JJ
   82 HAUT(JJ)=GGMIN+((RJJ*DGMAX)/10.)
C
C     CALCUL DE L ABCISSE DE L AXE DE LA RAIE POUR
C     LES NNIV NIVEAUX IMPOSES
      DO 53 I=1,NNIV
      DO 52 N=NNP,NND
      IF(GG(N).LE.HAUT(I))GO TO 1804                                SSSSSSS1
   52 CONTINUE                                                      SSSSSSS1
 1804 IF(GG(N).EQ.HAUT(I))GO TO 1807
      HDA=GG(N-1)-HAUT(I)
      HDB=HAUT(I)-GG(N)
      HDAB=GG(N-1)-GG(N)
      IF(HDAB.LT.PET1)GO TO 1805
      IF(HDA.LT.PET2)GO TO 1806
      IF(HDB.LT.PET2)GO TO 1807
      GO TO 54
 1805 SIGG5=SIGMA(N-1)+DELTAG/2.
      GO TO 1808
 1806 SIGG5=SIGMA(N-1)
      GO TO 1808
 1807 SIGG5=SIGMA(N)
      GO TO 1808
```

```
   54 SIGG5=SIGMA(N-1)+DELTAG*HDA/HDAB
 1808 NP1=N+1
C
      DO 55 N=NP1,NND
      IF(GG(N).GE.HAUT(I))GO TO 1814                                    SSSSS$S1
   55 CONTINUE                                                          SS$GSSS1
 1814 IF(GG(N).EQ.HAUT(I))GO TO 1817
      HDA=HAUT(I)-GG(N-1)
      HDB=GG(N)-HAUT(I)
      HDAB=GG(N)-GG(N-1)
      IF(HDAB.LT.PET1)GO TO 1815
      IF(HDA.LT.PET2)GO TO 1816
      IF(HDB.LT.PET2)GO TO 1817
      GO TO 56
 1815 SIGG6=SIGMA(N-1)+DELTAG/2.
      GO TO 60
 1816 SIGG6=SIGMA(N-1)
      GO TO 60
 1817 SIGG6=SIGMA(N)
      GO TO 60
   56 SIGG6=SIGMA(N)-DELTAG*HDB/HDAB
C
   60 SIGS(I)=(SIGG5+SIGG6)/2.
      SIGCD1(I)=SIGS(I)*(1.00+CC1)+CC2
      IF(I-1)83,83,84
   83 SIM(KJI+1)=SIGCD1(1)
      SIM(KJI)=LR
      IF(ILIST.EQ.0)GO TO 53
      DGMAX=DGMAX/GMAX
      PRINT87,LR,SIGGS(LJ),GGMIN,GGMAX,SIGS(I),SIGCD1(I),DGMAX
   87 FORMAT(/,1X,I5,F20.4,2F14.4,F20.6,F19.6,F20.4)
      GOTO53
   84 DSIGD1(I)=SIGCD1(I)-SIGCD1(1)
      SIM(KJI+3)=SIM(KJI+3)+DSIGD1(I)/8.
   53 CONTINUE
      IF(ILIST.EQ.0)GO TO 5353
      WRITE(6,89)(DSIGD1(I),I=2,9)
   89 FORMAT(8(5X,F10.6))
 5353 IF(KJI.EQ.397)GO TO 761                                           3/3/72
      KJI=KJI+4
      GO TO 762
  761 KJI=1
      IF(ILIST.EQ.0)PRINT 763,SIM
      IF(I17.EQ.1)WRITE(17)SIM
  762 CONTINUE
  763 FORMAT(1X,F7.0,F13.6,F8.0,F7.4)
      IF(ISURF.LE.0)GO TO 67
C
C     CALCUL DE LA LARGEUR EQUIVALENTE (FACULTATIF)
C
      DO 21 IS=1,ISURF
      DSIGS3=DSIGZ3(IS)
      NNGS3=DSIGS3/DELTAG
      RNNGS3=NNGS3
      ARIAS=SIGS(1)+RNNGS3*DELTAG-SIGIG
      RNNDS=(ARIAS/DELTAG)+1.
      NNDS=RNNDS
      NNPS=NNDS-(2*NNGS3)+1
      NTOTS2=(NNDS-NNPS)/2
      NSOMET=NNPS+NTOTS2
      GGMAXG=0.0
      DO 27 NN=NNPS,NSOMET
   27 IF(GGMAXG.LT.GG(NN))GGMAXG=GG(NN)
      GGMAXD=0.0
      DO 30 NN=NSOMET,NNDS
   30 IF(GGMAXD.LT.GG(NN))GGMAXD=GG(NN)
      BACKGR=(GGMAXD+GGMAXG)/2.
      LL=0
      DO 34 NN=NNPS,NNDS
      LL=LL+1
   34 G(LL)=BACKGR-GG(NN)
      NTOT=(NTOTS2*2)+1
      G(1)=G(1)/2
      G(NTOT)=G(NTOT)/2
      SURF=0
      DO 35 NN=1,NTOT
   35 SURF=SURF+G(NN)
      ARGEQ=(SURF*DELTAG)/BACKGR
      DSIGS4=DSIGS3*2
   21 PRINT 92,DSIGS4,BACKGR,ARGEQ
   92 FORMAT(3X,E15.8,7X,E15.8,9X,E15.8)
C
   67 CONTINUE
C
  195 DO 16 I=1,1024,4
      GG(I)=GG(I+5120)
      GG(I+1)=GG(I+5121)
      GG(I+2)=GG(I+5122)
   16 GG(I+3)=GG(I+5123)
      FKP=KP
      SIGIG=SIGIG1+FKP*BENGTH
      SIGA=SIGB+DELTAG/100.
      SIGB=SIGIG+BENGTH+GARDE
      IF(KP.EQ.NB-1)SIGB=SIGLIM
C
  192 CONTINUE
C
C
      IF(KJI.EQ.1)GO TO 301
      DO 764 I=KJI,400
  764 SIM(I)=0.
      IF(ILIST.EQ.0)PRINT 763,SIM
      IF(I17.EQ.1)WRITE(17)SIM
      GO TO 301
 6111 PRINT 6068,NRAIE,NSIMAX
 6068 FORMAT(/,2X,I8,' RAIES DANS UN BLOC  (NOMBRE MAX.=',I7,')',/)
  301 PRINT 201,LR
  201 FORMAT(/,'   FIN DU TRAVAIL',I7,' RAIES TROUVEES',///)             3/3/72
C
C    FERMETURE DU FICHIER BENSON.
      IF(IPLO.EQ.0)GO TO 22
      NETUD=-900-IFICH
      CALL PNUMA(DSIGMA,0.,NETUD,0.,0.)
      IFL=NB/3
      IF(IFL*3.LT.NB)IFL=IFL+1
      FL=IFL*(DSIGMA+2.)/100.
      PRINT 202,FL
  202 FORMAT(/,'   TRACE TERMINE  (',F6.1,' METRES.)')
C
   22 STOP
      END
```

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

CENTRE INTER-REGIONAL DE CALCUL ELECTRONIQUE

MANUEL

DU

PARFAIT PETIT FOURIERISTE

par J. CONNES
H. DELOUIS
F. GAUTIER

Version 1.4

TABLE DES MATIERES

GENERALITES — pages

I.- Généralités 1

II.- Notations 2

III.- Tracés 3

IV.- Normalisation, Apodisation, Interpolation du spectre 4

V.- Différentes bandes et leurs caractéristiques 5

VI.- Représentation et organisation des différents types de bandes 7

VII.- Cartes de contrôle 8

LES DIFFERENTS PROGRAMMES ET LEUR MODE D'EMPLOI

I.- DUMP DE BANDE
1.- Programme "DUMPBDE" 10
2.- "DSP" "TD" (TAPE DUMP) 12

II.- Programmes "TEST8" et "TEST8S" 13

III.- Programme "BSS10" 21

IV.- Programmes "TFID3" et "T102" 26

V.- Programme "MPLOT" et "MPLØS" 29

VI.- Copie de fichiers bande
1.- Programme "UTICOPIE" 31
2.- Programme "B5COPIE" 32
3.- Programme "C6COPIE" 33

APPENDICE A

I.- Création d'une bande au CIRCE 34

II.- Demande de trace au dispatching 35

III.- Note remise au dispatching en cas de cancel d'un job 36

APPENDICE B

I. - Dump d'une bande par "DUMPBDE" "TAPE" 7 pistes 37

II. - Décodage d'un DUMP HEXA 38

III. - Analyse des "SENSE BYTES" 40

APPENDICE C

Tape Dump (DSP "TD") 44

APPENDICE D

- Note à propos des programmes TEST8 et BSS10 et liste des programmes distribués 47

- Exemples de durée pour TEST8 et BSS10 47

APPENDICE E

Extrait du "PROGRAMMER'S GUIDE" à propos des fichiers Fortran 48

APPENDICE F

Dictionnaire des "TRUFFES" les plus courantes 56

APPENDICE G

Utilisation de la bibliothèque des programmes. Liste des cartes contrôle 59

APPENDICE H

Exécutions successives d'un même programme en un seul job 67

APPENDICE I

Particularités des programmes pour Mr CONNES

- Exemples de durée. 69
- Le codage des "TAPES" 9 pistes 70
- Le programme "ANABRUINT" 72

APPENDICE J

Le paramètre "IAP" et les différentes apodisations possibles 73

Pour exploiter les programmes qui lui sont remis, l'utilisateur doit remplir deux sortes de cartes :

- les cartes données qui donnent les caractéristiques de l'interférogramme et du spectre qu'on veut calculer et dont le format a été imposé par le programmeur ,

- les cartes contrôle nécessaires à l'exécution du programme et à la lecture ou la création de tous les fichiers et dont le format est imposé par le système d'exploitation.

G E N E R A L I T E S

oOo

I - GENERALITES

Un interférogramme ayant été enregistré sur une bande que dans toute la suite nous appellerons "Tape", on peut vouloir :

- ou bien simplement lire la "Tape" et la transformer en une bande binaire "FT12" (exploitable ultérieurement) avec le programme TEST8S ;
- ou bien tester la "Tape" en ayant le décodage des blocs erronés, avec le programme DUMPBDE. La liste obtenue est une image de la configuration de bits se trouvant réellement sur la bande ;
- ou bien, si la bande "Tape" est bonne ou supposée telle, il faut tester la valeur de l'interférogramme lui-même. Si son nombre "M" de valeurs est petit ($M < 32768$), le spectre peut être calculé directement et entièrement par le programme "BSS10" qui utilise un programme F.F.T. (Fast Fourier Transform) entièrement en mémoire centrale de l'ordinateur.

S'il a beaucoup plus de valeurs, on le teste à basse résolution avec le programme "BSS10" et à haute résolution avec le programme "TEST8" qui permet le calcul d'une tranche étroite du spectre à la résolution maximum. Dans ce dernier programme aussi, le calcul est fait par un programme F.F.T. entièrement en mémoire centrale. On a le choix de la largeur et de la position de la tranche calculée, ce qui permet d'évaluer le rapport s./b. à partir de deux tranches convenablement choisies dans le spectre.

Après le passage de ces programmes de test, l'exploitation proprement dite d'un grand interférogramme se fait avec d'autres programmes :

- Le programme "TFID3" permet de calculer la T.F. d'interférogrammes avec une valeur de "M" $\leq 2.10^6$. La T.F. ne pouvant plus être réalisée directement et globalement en mémoire centrale, on se ramène à des calculs partiels en mémoire centrale, les résultats intermédiaires et le spectre final étant stockés sur disque magnétique.

- Le programme "T102" traite les valeurs B du spectre calculé par "TFID3" et stocké sur disque. Il les normalise et les recopie sur une bande "FT16", puis il trace automatiquement la région où la normalisation a lieu.

Ces deux derniers programmes peuvent être passés séparément ou regroupés en deux étapes successives d'un même "JOB". Toutes les exploitations ultérieures du spectre ainsi calculé se feront à partir de la "FT16" ainsi créée.

..

- Le programme "MPLØT" est uniquement chargé d'interpoler et de tracer le spectre.

- Le programme "DERPTE" permet le repérage et le pointé automatique des raies du spectre.

II - QUELQUES NOTATIONS

- h = "DELTAD" = pas de l'interférogramme en cm.
- "FRANGE" = unité de base des pas h en cm. ((*))
- "IFR" ⟷ h/FRANGE.
- M = nombre de valeurs de l'interférogramme.
- "TAPE" Bande Interf. enregistrée (7 ou 9 cnx)
- "FT12" " " binaire ("FT32" parfois) (9 cnx)
- "FT16" " Spectre résultat de la TF (valeurs) (9 cnx)
- "FT14" " " (tracé) (9 cnx)

Signification des paramètres

- ICHEN1	:	1	on fait les calculs en partant de la lecture de la "tape".
		0	on fait les calculs en partant de la lecture de la FT12.
- IC6	:	1	on crée une FT16.
		0	on ne crée par de FT16.
- IAP	:	1	on apodise. (voir appendice J)
		0	on n'apodise pas.
- IMP ou IIMP	:	Test permettant de sortir plus ou moins de résultats sur le "listing" (voir chaque programme particulier).	

((*)) FRANGE = +0.3507984664 D-05
+0.79123932 D-05

..

- SIGB : valeur située dans l'intervalle spectral ou dans la tranche que l'on veut calculer.

- M : nombre de points de l'interférogramme.

- IFR : nombre de franges élémentaires par pas de l'interférogr.

- NU : numéro de l'interférogramme.

- SIG1 : abcisses approximatives entre lesquelles on veut
- SIG2 : tracé du spectre ; si on veut tout l'intervalle spectral, on prend SIG1 très petit et SIG2 très grand.

- NBLOC : nombre de blocs de 1024 valeurs qu'on veut tracer sur une même feuille.

- IPLO :
  - 1 on trace les valeurs à échelle normale,
  - 9 on trace les valeurs à échelle normale et dilaté en amplitude,
  - 8 on fait uniquement le tracé dilaté.

- IR : indique si on veut calculer la première moitié ou la deuxième moitié de l'intervalle spectral (TFID).

III - GENERALITES SUR LES TRACES

- "DX" : échelle en cm^{-1} par cm de papier en abcisse.

- "IPLØ" :
  - 1 tracé normal
  - 9 tracé normal et dilaté par 10,
  - 8 dilaté par 10 seul.

- "IFICH" : numéro du fichier sur la bande Benson.

- "HT" : hauteur totale du tracé en cm (0<HT≤30 cm - pt format) (0<HT≤70 cm - gd format)

- "TEXT" : texte perforé sur une carte donnée qui sortira en tête du tracé pour le caractériser.

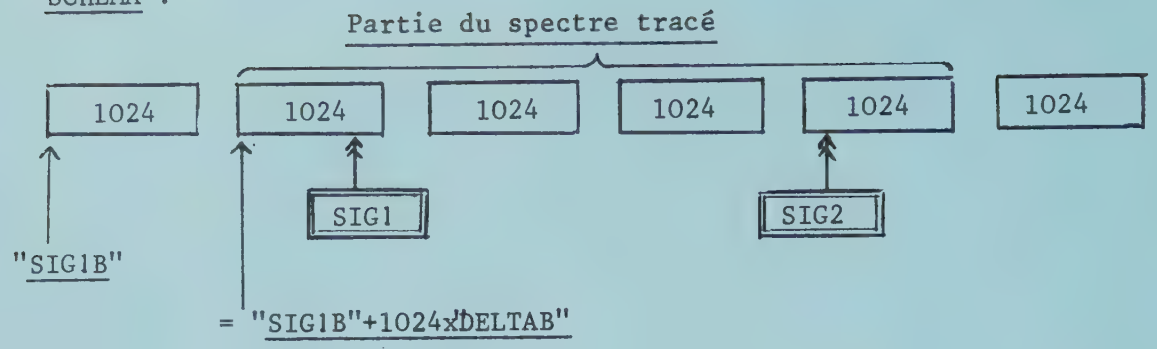
- Limites du tracé :

Après calcul de la T.F., les valeurs du spectre B, ou points primaires, sont réparties dans un ou plusieurs blocs de 1024 valeurs. Le programme qui trace le spectre laisse le choix du premier et du dernier bloc à tracer, mais en aucun cas(*) on ne peut commencer ou terminer le tracé au milieu d'un bloc de 1024 valeurs. Le choix des blocs de début et de fin de tracé se fait en donnant deux abcisses "SIG1" et "SIG2" en cm^{-1}. Le spectre se fera alors du bloc où se trouve la valeur du spectre correspondant à l'abcisse "SIG1" à celui où se trouve celle correspondant à l'abcisse "SIG2".

(*) Seul le programme "MPLØS" ne respecte pas ces conventions.

(voir p. 29).

SCHEMA :



L'abcisse du premier échantillon du spectre B est égale à "SIG1B" et le pas d'échantillonnage est "DELTAB" (on retrouve ces valeurs avec ces notations dans le"listing" des programmes).

Le tracé des blocs choisis se fera par pages séparées de "NBLØC" blocs de 1024 valeurs, chaque page ayant son système d'axes. La longueur en CM^{-1} d'un bloc est listée sous la valeur de "BENGTH". C'est la longueur minimum d'une page si NBLØC=1

(Voir modèle d'une demande de tracé au dispatching dans l'appendice A (p. 35)).

IV - NORMALISATION - APODISATION - INTERPOLATION DU SPECTRE

Les valeurs du spectre B obtenues après calcul de la T.F. sont normalisées en ramenant la plus grande valeur positive à l'unité. Dans le cas d'un spectre déformé ou inversé, on pourra donc avoir des valeurs négatives énormes.

Ces échantillons B ne sont pas assez serrés pour être tracés directement, ce qui nécessite l'interpolation de chaque bloc de 1024 valeurs à tracer. On passe alors des valeurs primaires B,(SIG1B,DELTAB) aux valeurs interpolées G,(SIG1G,DELTAG) avec :

DELTAG = DELTAB/5.

On a donc un échantillonnage G cinq fois plus serré.

C'est au cours de cette interpolation qu'on peut pondérer l'interférogramme. Dans ce cas, au lieu de réaliser l'interpolation en convoluant le spectre B par une fonction sin x/x classique, on le convolue par la T.F. de la fonction qui aurait pondéré directement l'interférogramme. Cette méthode nous amène à faire deux remarques :

1/ La normalisation réalisée sur les valeurs primaires B n'est plus qu'approximative au niveau des valeurs interpolées G (en général G > B) ;

2/ Les valeurs primaires gardées sur les bandes "FT16" correspondent toujours à la T.F. d'un interférogramme pondéré Ce n'est qu'au moment de l'exploitation de cette bande (tracé, pointé de raies, moyenne, etc...) qu'on se pose le problème de l'apodisation ou de la réduction du pouvoir de résolution.

..

DIFFERENTES BANDES ET LEURS CARACTERISTIQUES

I. - GENERALITES, DEFINITIONS

LRECL : logical record length
BLKSIZE ; block size, taille d'un bloc physique sur la bande, c'est-à-dire entre 2 gaps.
VS : variable "spanné"
SL : standard label
NL : No Label

Une bande "standard label" est une bande qui comporte avant l'écriture des informations composant le 1er fichier, un fichier écrit par le système (composé de 3 enregistrements de 80 caractères chacun). Ces enregistrements donnent des indications sur le numéro de la bande, le job qui a créé le fichier qui suit, les techniques d'écriture sur la bande, etc...

Ce fichier label est séparé du fichier utilisateur par une tape mark (EOF ou End of File).

Le fichier utilisateur est suivi (séparé de lui par un EOF) d'un fichier label (2 enregistrements de 80 caractères chacun). Ensuite, sur la bande, chaque fichier est entouré d'un label de tête et d'un label de queue. Il n'y a que le premier fichier dont le label de tête comporte un enregistrement supplémentaire donnant le numéro de la bande.

Une bande NO LABEL (sans label) est une bande qui commence immédiatement par le fichier utilisateur. Et tout au long de la bande, chaque fichier utilisateur est séparé du précédent et du suivant par un EOF. (pour plus de détails, voir la brochure IBM TAPE LABELS ref. C28-6680).

En fortran, si la taille du block size est inférieure à la longueur de la liste (c'est-à-dire ce qu'on veut lire ou écrire sur la bande en une seule fois) on doit éclater l'information sur plusieurs blocs, c'est la technique du spanning (voir photocopie des pages 59-64 du "Programmer's Guide Fortran"). Le format correspondant figurant dans les cartes DD est RECFM=VS. RECFM étant l'abréviation de record format.

C5 ou TAPE : bande interférogramme telle qu'elle sort de l'enregistreur de laboratoire .(voir appendice I pour les bandes 9 CNX p. 69), 7 canaux, 556 bpi, BCD, blocksize : 600 octets, No Label (NL) "spanné" (TRTCH=ET)

B5 ou FT12 : bande image de la Tape, 9 cx, 1600 bpi, binaire, blocksize : 1008, LRECL : 1004, variable "spanné", labellé (SL)

Les enregistrements logiques de la FT12 sont de 10 000 mots (de 4 octets) éclatés sur des blocs physiques de 1000 octets (+ 4 octets, mot de contrôle pour l'enregistrement, + 4 octets, mot de contrôle pour le bloc).

Seul le premier est différent, puisqu'il a une longueur de 32 octets (24 de texte + 8 octets de contrôle). Ce premier enregistrement contient le titre NU,M,NU,M,NU,M.

C6 ou FT16 : bande résultat contenant les valeurs primaires du spectre avant interpolation et avant apodisation.

9 cx, 1600 bpi, binaire, blocksize : 1032, LRECL 1028, variable "spanné", labellé (SL).

Les enregistrements logiques de la FT16 sont de 1024 mots (de 4 octets) éclatés sur des blocs physiques de 1024 octets (+ 4 octets, mot de contrôle pour l'enregistrement, + 4 octets, mot de contrôle pour le bloc).

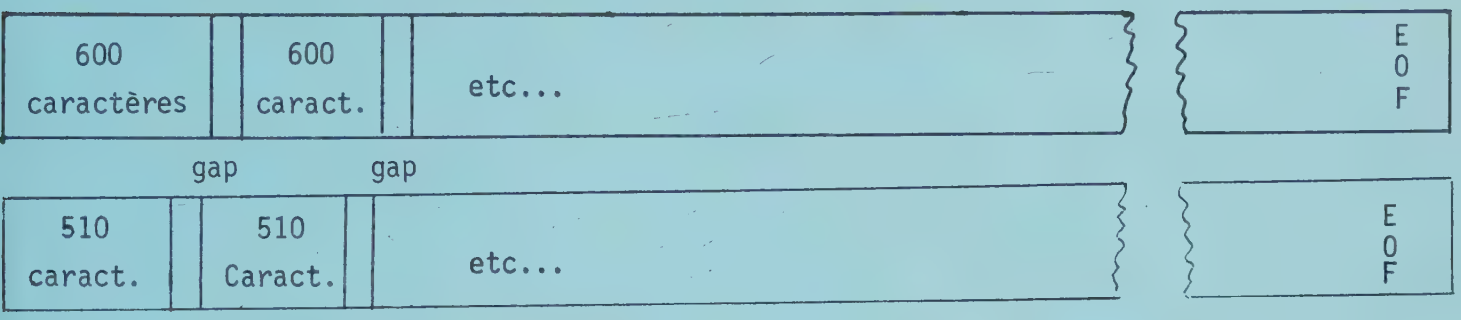
Là aussi le premier enregistrement est plus court. Il a pour longueur 40 octets (32 de texte + 8 de contrôle). Il contient le titre NU,NU,NU,ND1024,SIG1B,DELTAB.

FT14 : BANDE PLOT. Contient les ordres traceur pour le Benson.

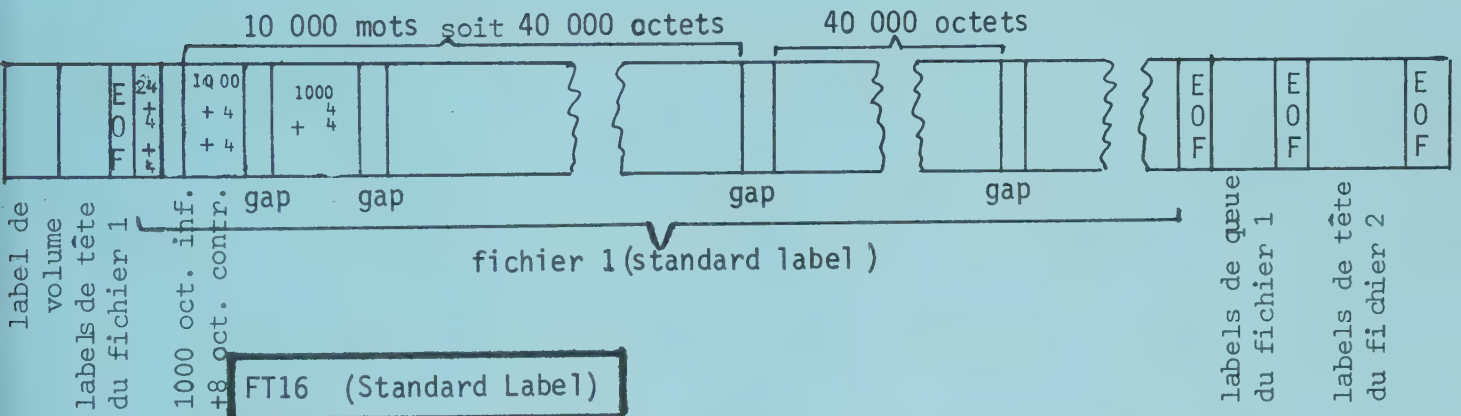
9 cx, 800 bpi, blocksize = 4000,
format = F labellé (SL).

REPRESENTATION ET ORGANISATION DES DIFFERENTS TYPES DE BANDES

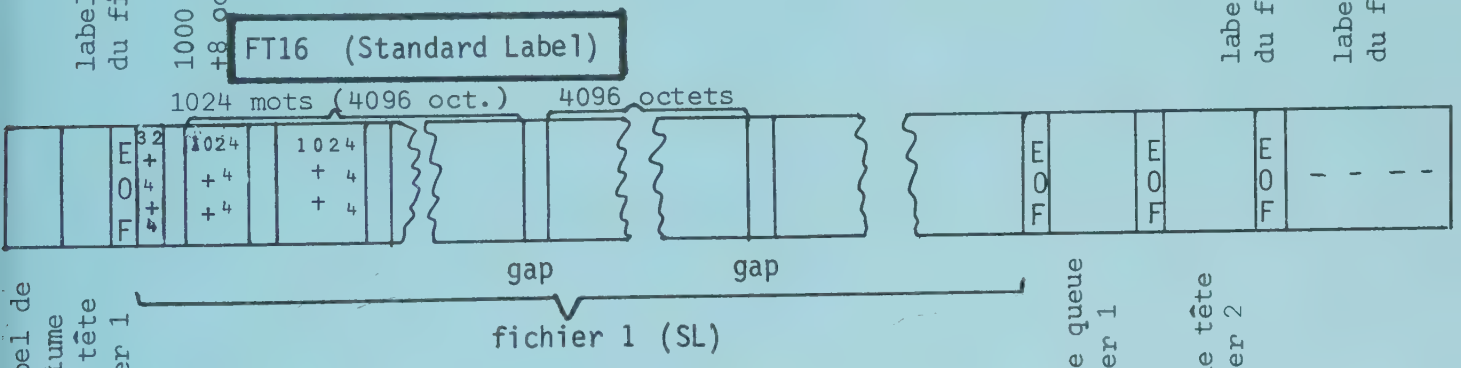
C5 (Sans Label)



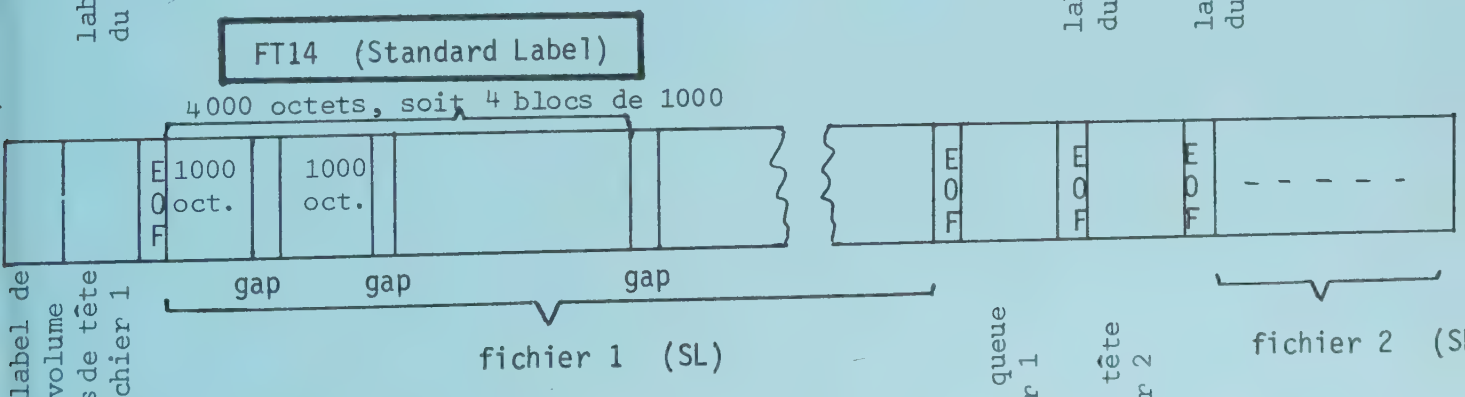
FT12 (Standard Label)



FT16 (Standard Label)



FT14 (Standard Label)



REVUE GENERALE DES CARTES DE CONTROLE NECESSAIRES POUR L'UTILISATION DES PROGRAMMES

Seules les cartes susceptibles d'être modifiées par l'utilisateur sont indiquées.

CARTE JOB

```
//BSS10 ß JØB ß CØT0830,TIME=(t,s),GUELACH
/*MAIN ß LINES=1
```

ß : blanc obligatoire

t = temps CPU en minutes (s en secondes)
l = nombre de lignes en milliers

CARTE SETUP

(nécessaire pour indiquer à l'opérateur (via ASP) le numéro de la bande à monter)

pour C5 (TAPE)

```
/*SETUP ß UNIT={BDE7 / BD16},DDNAME=TAPE,ID=(009245,SAVE,NØRING,NL)
```

pour B5 (FT 12)

```
/*SETUP ß UNIT=BD16,DDNAME=FT12F001,ID=(103274,SAVE,RING,SL)
```

pour FT16

```
/*SETUP ß UNIT=BD16,DDNAME=FT16F001,ID=(103275,SAVE,RING,SL)
```

pour FT14

```
/*SETUP ß UNIT=BD16,DDNAME=FT14F001,ID=(104262,SAVE,RING,SL)
```

CARTE DD

(nécessaire pour indiquer à l'OS (Operating System) les caractéristiques de la bande utilisée)

pour C5 ou TAPE

```
//GØ.TAPE ß DD ß UNIT=BDE7,LABEL=(n,NL),DISP=(ØLD,KEEP),
// ß DCB=(TRTCH=ET,DEN=1,BUFNØ=1)
```

```
//GØ.TAPE ß DD ß UNIT=BD16,LABEL=(n,NL),DISP=(ØLD,KEEP),
// ß DCB=(DEN=2,BUFNØ=1)
```

pour FT12

```
//GØ.FT12F001 ß DD ß UNIT=BD16,DSN=FT12812,LABEL=(n,SL),
// ß DCB=(RECFM=VS,LRECL=1004,BLKSIZE=1008)
```

pour FT16

```
//GØ.FT16FOO1 ⊬ DD ⊬ UNIT=BD16,DSN=FT16812,LABEL=(n,SL),
// ⊬ DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032)
```

pour FT14

```
//GØ.FT14FOO1 ⊬ DD ⊬ UNIT=BD16,LABEL=(n,SL),
// ⊬ DCB=(RECFM=F,BLKSIZE=4000,DEN=2)
```

La carte SETUP se met immédiatement derrière la carte JØB.

Les cartes DD se mettent entre la fin du programme et les données. Voir les exemples particuliers pour chaque job.

REMARQUE sur les cartes SETUP et DD :

Une bande dont le numéro est indiqué en SETUP reste montée tout le temps du job, sauf si on donne au système l'ordre d'en monter une autre à sa place. Il suffit pour cela de ne pas mettre de carte SETUP pour la 2ème bande, d'indiquer le numéro de cette dernière dans la carte DD et de mettre un paramètre spécial pour indiquer au système qu'il doit monter les bandes sur le dérouleur où était la première.

EXEMPLE : Job utilisant une TAPE en entrée et une FT14 en sortie :

```
//    carte JØB
/*SETUP ⊬ UNIT=BD16,DDNAME=TAPE,ID=(009245,SAVE,NØRING,NL)}  carte SETUP
                                                              correspondant à
  ≀                                                           la 1ère bande montée
  ≀ programme
/*
//GØ.TAPE ⊬ DD ⊬ UNIT=BD16,LABEL=(n,NL),}   carte DD correspondant à la
                                            1ère bande montée.
// ⊬ DCB=(DEN=2,BUFNØ=1)
//GØ.FT14FOO1  UNIT=AFF=TAPE,LABEL=(n,NL),} carte DD correspondant à la
                                            2ème bande montée.
// ⊬ DCB=(RECFM=F,BLKSIZE=4000,DEN=2),
// ⊬ VØL=SER=103328
```

Cette disposition implique qu'il n'est plus fait appel au fichier "tape" dans le programme en train de s'exécuter une fois que la 1ère utilisation du fichier "FT14" a provoqué le démontage de la 1ère et le montage à sa place de la 2ème bande.

Le paramètre "AFF=" n'est pas nécessaire s'il s'agit de deux étapes d'un même job comme dans le cas du TFID + T102.

DUMP (*) DE BANDE

- Programme "DUMPBDE"
- "DSP" "TD" (TAPE DUMP)

(*) liste de caractères représentant la configuration des bits qui se trouvent sur un support d'information tel que bande magnétique ou mémoire centrale par exemple.

PROGRAMME "DUMPBDE"(*)

UTILISATION DU PROGRAMME DUMPBDE

C'est le paramètre PARM de la carte EXEC qui permet d'indiquer au programme DUMP les options choisies.

PARM='XXXXXX,HEXA,XXX,BAD,XXXXXX'

XXXXXX — Un nombre inférieur ou égal à 6 chiffres, permet d'indiquer le nombre de blocs désirés.
OPTION PAR DEFAUT = toute la bande.

HEXA — Si l'on désire la représentation HEXADECIMALE de la bande
OPTION PAR DEFAUT = EBCDIC

XXX — Un nombre inférieur ou égal à 3 chiffres pour indiquer le nombre maximum de blocs incorrects consécutifs admis.
OPTION PAR DEFAUT : 10

BAD — Si l'on désire la liste des seuls enregistrements incorrects.

XXXXXX — Un nombre de 6 chiffres au plus pour indiquer le nombre de blocs à sauter.
OPTION PAR DEFAUT = 0

EXEMPLES : PARM='100,HEXA,,,10'

Je désire lire 100 blocs consécutifs à partir de l'enregistrement 11 compris, la représentation sera en HEXA :

PARM=',,,,3000'

Je désire lire toute la bande à partir de l'enregistrement 3001 compris, la représentation sera en EBCDIC :

PARM=',HEXA,,BAD'

Seuls les enregistrements PHYSIQUEMENT incorrects seront listés, la représentation sera en HEXA.

* programme fourni par IBM (écrit en assembleur).

EXEMPLE D'UTILISATION DU PROGRAMME DUMP.

```
//DELDUMP JOB CIR0876,DUPPONT,TIME=(0,18),MSGLEVEL=(0,0)
//*          DUMP DE TAPE 9 CNX.
//*         **********************************
/*SETUP UNIT=BD16,DDNAME=DDTAPE,ID=(009593,NORING,SAVE,BLP)
//JOBLIB DD DSN=BP1.LINKLIB,DISP=SHR
// EXEC PGM=DUMPBAND,PARM='10,HEXA,,,10'
//GO.DDTAPE DD UNIT=BD16,LABEL=(01,NL),DISP=(OLD,KEEP),
// DCB=(RECFM=U,BLKSIZE=8672,BUFNO=1,DEN=2)
//GO.DDIMPR DD SYSOUT=A
//GO.SYSUDUMP DD SYSOUT=A
/*
//
```

REMARQUE : (n,SL)≡(n',NL) avec n'=3n-1

Codage d'une "TAPE" 7 pistes ⟶ (appendice B p. 37)

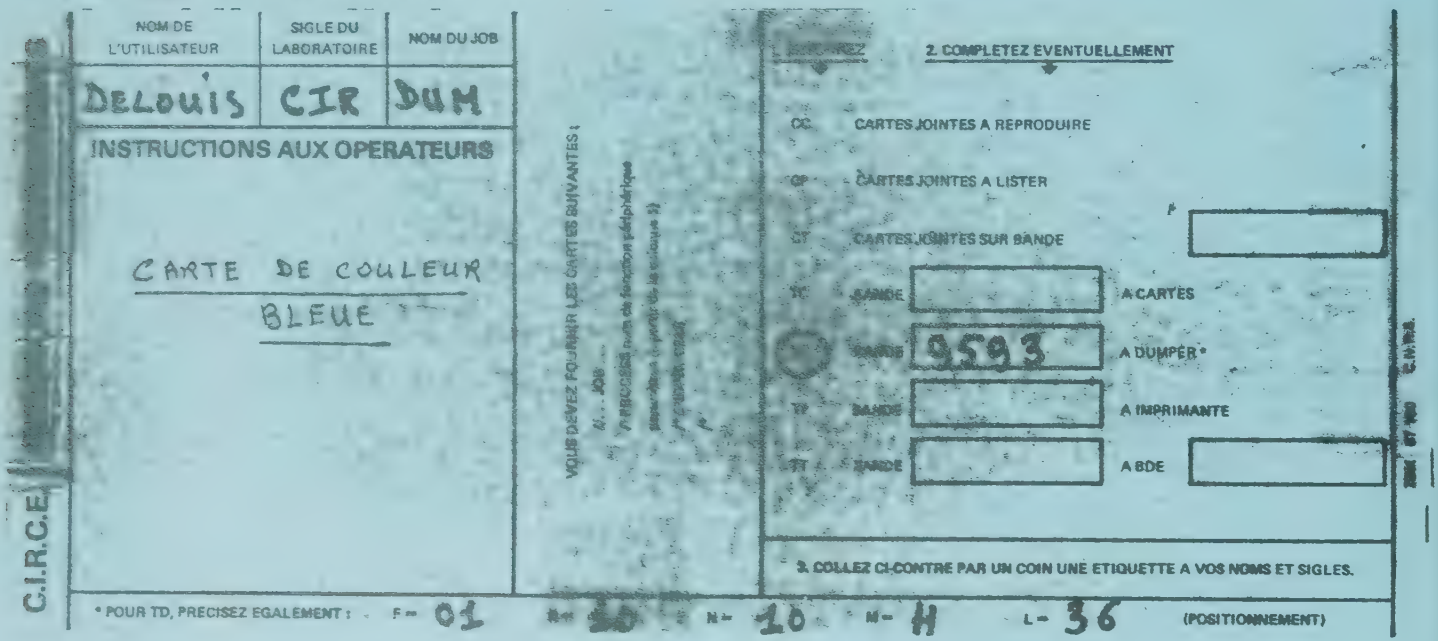
Codage d'une "TAPE" 9 pistes ⟶ (appendice I p. 70)

DUMP D'UNE BANDE PAR UN 'DSP'

Le système "ASP" contient un certain nombre de programmes (DSP) (Dynamic Support Programs) qui mettent en oeuvre des opérations périphérique annexes telles que le dump de bande appelé "TD".

Ci-dessous, un exemple de cartes de contrôle permettant le dump d'une bande 9 pistes de type "TAPE" (bande enregistreur). Ce sont ces blocs 11 à 20 du deuxième fichier qui seront dumpés.

```
//
/*
/*ENDPROCESS
IN=BD16,OUT=1403,ID=009593,DEN=8,MD=0
/*PROCESS TD
//DUM JOB CIR0819,DELOUIS
```



On trouvera dans l'appendice C (p.44) tous les détails d'utilisation de ce DSP.

REMARQUE : La densité ici n'est pas codée comme dans le paramètre "DEN" du DCB d'une carte DD.

Pour les "TAPES" 7 pistes, il faut prendre :

IN=BDE7,DEN=5,MD=ET

P R O G R A M M E "TEST8 S" et "TEST8"

- GENERALITES
- CARTES DE CONTROLE
- CARTES DONNEES

PROGRAMME "TEST8" - GENERALITES(*)

Principe du calcul d'une tranche spectrale k fois plus étroite que le spectre normal.

Soit I_1 un interférogramme de M_1 valeurs avec $M_1 \leq 2^{n1}$ et soit h_1 son pas d'échantillonnage. Son spectre B_1 aura une largeur $\Delta\sigma_1 = 1/2\ h_1$.

Le programme "TEST8" calcule à partir de I_1 un nouvel interférogramme I_2, de pas h_2, k fois plus large, c'est-à-dire avec seulement $M_2 = M_1/k$ points. Le facteur de réduction k sera toujours un nombre entier.

$$M_2 < 2^{n2}$$

Cet interférogramme I_2 aura le même spectre B_2 que I_1, mais son domaine sera k fois plus petit et égal à :

$$\Delta\sigma_2 = \Delta\sigma_1/k = 1/2\ h_2.$$

Pour l'exécution du programme, on ne fixe pas le facteur de réduction k mais la valeur de M_1 et la valeur de 2^{n2} ="NPRIM" désirée.

D'où : $k \geq \dfrac{M1}{0.9 * NPRIM}$

Pour un programme "TEST8" donné, la valeur de "NPRIM" est fixée et elle est indiquée sur la tranche du paquet de cartes correspondant. Le choix de "NPRIM" se ramène donc au choix du programme pour lequel "NPRIM" a la valeur désirée (ou qui s'en rapproche le plus).

Une fois le calcul de I_2 fait, la T.F. se fait avec "NPRIM" valeurs en mémoire centrale. "NPRIM" fixe donc le nombre de points de sortie du spectre. Il existe des programmes permettant le calcul de tranches de 1024, 2048, ... 32768 valeurs.

Parmi les k tranches découpées dans le spectre initial B_1, le programme calcule celle qui contient la valeur du spectre correspondant à l'abcisse "SIGB" indiquée en cm^{-1} dans les cartes données. Il est important de remarquer que si on a le choix de la tranche, on n'a pas le choix de ses limites puisque, pour une valeur donnée de k, le découpage en k tranches égales du spectre initial est unique.

Pour des raisons de filtrage, les bords de la tranche spectrale calculée sont déformés, ce qui implique de ne pas prendre une valeur de k trop élevée si on veut que le centre de cette tranche ne soit pas distordu (en général, prendre $k < 32$).

(*) H. DELOUIS - Thèse 3ème cycle, 1968, chapitre II p.50.

..

Cas particulier d'utilisation :

Si $M_1 < NPRIM$, on aura k = 1 et le spectre calculé sera le même que celui qu'on obtiendrait en utilisant un programme "BSS1O" (Le temps de calcul sera plus long et ce n'est donc pas la bonne méthode pour un calcul complet à basse résolution).

Si $M_1 \ll NPRIM$, le spectre sera même déformé, car l'interpolation finale du spectre n'a pas été prévue pour ce cas particulier d'utilisation par ailleurs très peu intéressant.

Remarque :

On trouvera dans l'appendice D une note concernant les programmes distribués et l'estimation de leurs caractéristiques (encombrement, durée, nombre de lignes).

Remarque :

La technique d'"overlay" est utilisée pour que les deux parties de ce programme (SP1 et SP5) ne soient pas chargées en même temps. Dans le fichier du "link-edit", il faudra donc inclure les cartes suivantes (ou vérifier qu'elles y sont bien).

(*)
```
Insert PP
Overlay alpha
Insert SP1,PLECB2,FLAG,PLASS
Overlay alpha
Insert SP5,FFT128
```

- Au moment de la compilation la carte EXEC aura la forme suivante :

```
// EXEC  FTHCLG,PARM.FØRT='NAME=PP,DECK'
PARM.LKED='OVLY,LIST,MAP'
```

(*) Les noms des sous-programmes chargés peuvent changer d'une version à l'autre.

CARTES DE CONTROLE

TEST8 A PARTIR DE LA "TAPE"

```
//TEST8 JOB COT0830,MAILLARD,TIME=(0,30),MSGLEVEL=(0,0)
/*SETUP UNIT=BDE7,DDNAME=TAPE,ID=(009241,NORING,SAVE,NL)
/*SETUP UNIT=BD16,DDNAME=FT12F001,ID=(103358,RING,SAVE,SL)
/*SETUP UNIT=BD16,DDNAME=FT14F001,ID=103182
//ETAP1 EXEC   FTHLG,PARM.LKED='OVLY',REGION.GO={150K / 215K / 450K}
//LKED.SYSIN DD *
    .....
     .....
     'DECK' DU TEST8. { 4K / 16K / 64K }
      .....
       .....
/*
//GO.TAPE DD   UNIT=BDE7,DISP=(OLD,KEEP),DCB=(DEN=1,TRTCH=ET,BUFNO=1),
// LABEL=(2,NL)
//GO.FT12F001 DD UNIT=BD16,DISP=(NEW,KEEP),LABEL=(10,SL),DSN=FT122033,
// DCB=(BUFNO=2,RECFM=VS,LRECL=1004,BLKSIZE=1008)
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2)
//GO.IMP DD SYSOUT=A
//GO.SYSIN DD *
   LUMIERE BLANCHE  PBS2 WARM,WHITH GLUE                    11006
 2033     85200   12+0.45000000D+04
+0.39000000D+04+0.60000000D+04   16    1     1
+0.10000000D+02+0.200E+02 2033 NEW PBS2 WARM ,WHITH GLUE
/*
//
```

Si les programmes sont en bibliothèque sur disque magnétique, voir en complément l'appendice G p.59.

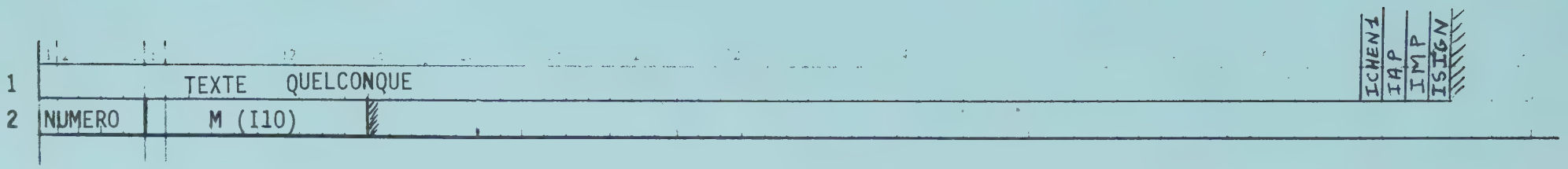
TEST8 A PARTIR DE LA FT12

```
//TEST8 JOB COT0830,GUELACH,TIME=(,20),MSGLEVEL=(0,0)
/*SETUP UNIT=BD16,DDNAME=FT12F001,ID=(104328,NORING,SAVE,SL)
/*SETUP UNIT=BD16,DDNAME=FT14F001,ID=103182
//ETAP1 EXEC   FTHLG,PARM.LKED='OVLY',REGION.GO={130K / 190K / 430K}
//LKED.SYSIN DD *
     .....
      .....          { 4K
   'DECK' DU TEST8   { 16K
        .....        { 64K
         .....
/*
//GO.FT12F001 DD DSN=F123031,UNIT=BD16,DISP=OLD,LABEL=(01,SL),
// DCB=(BUFNO=2,RECFM=VS,LRECL=1004,BLKSIZE=1008)
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2)
//GO.IMP DD SYSOUT=A
//GO.SYSIN DD *
                                                          1100
 3031    140000   11+0.77050000D+04
+0.76800000D+04+0.77300000D+04   16    9    1
+0.20000000D+00+0.250E+02  3031 PU239 21 06 71 M 140000
/*
//
```

- Dans ce programme, le sous-programme "SP1" ainsi que tous les programmes de lecture de la "TAPE" sont supprimés.

- Si les programmes sont en bibliothèque sur disque magnétique, voir en complément l'appendice G p. 59.

TEST8S - CAS GENERAL



- Pour la signification des variables et tests, voir le TEST8 page suivante.

- Ce programme permet uniquement de constituer une FT12 à partir d'une C5 ("TAPE").

- REGION.GØ ≃ 86 K.

- Durée : pour 10.000 valeurs décodées et recopiées, prévoir { 0,88 s / 0,30 s } pour une "TAPE" { 7 pistes / 9 pistes }

TEST8 - CAS GENERAL

1		TEXTE QCQ						ICHEN1 IAP IIMP ISIGN
2	NUMERO	M (I10)	IFR	SIGB D15.8				
3	SIG1	(D15.8)	SIG2 (D15.8)		NBLØC	IPLØ	IFICH	
4	DX	(D15.8)	HT (E10.3)		TEXT qui sortira sur le tracé col 72			

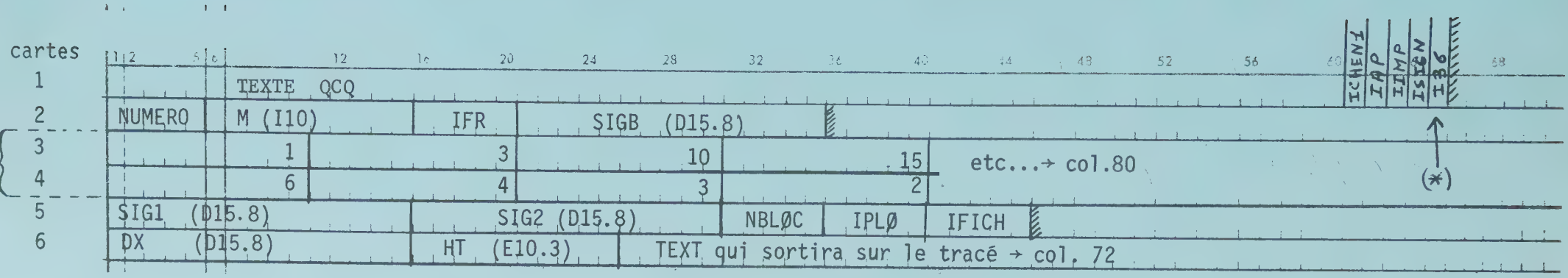
- ICHEN1 { = 0 : Lecture de l'interf. sur une bande binaire 9 pistes (SL) "FT12".
  { > 0 : Lecture de l'interf. sur une bande enregistrée (NL) "TAPE" et création de la FT12.
- IAP : Si ce test est > 0 il y a apodisation au niveau de l'interpolation du spectre.

Si ICHEN1 > 0 {
- IIMP { = 0, seules les 10 premières valeurs de chaque bloc de 10.000 valeurs sont listées (sauf TEST8S).
  { = 1, tous les blocs de 100 valeurs lus, puis décodés sur la "TAPE" sont listés par PLECB2 (6 ou 3).
  { = 2, les 10 premières valeurs de chaque bloc de 100 valeurs sont listées.
- ISIGN ≠ 0, tous les échantillons de l'interf. sont changés de signe.

- SIGB : Abscisse quelconque en CM^{-1} de la tranche désirée.
- NUMERO : De l'interf.
- M : Nombre d'échantillons à lire et à garder pour le calcul.
- IFR : Nombre de franges élémentaires de référence par pas.
- DX : Echelle du tracé en CM^{-1}/CM de papier sur l'axe des X.
- HT : Hauteur du tracé en CM.
- SIG1, SIG2 : Limites du tracé. Si on veut toute la tranche prendre "SIG1" très petit et "SIG2" très grand.
- IPLØ { =1 : Tracé normal.
  { =9 : Tracé normal + tracé dilaté par 10.
  { =8 : Tracé dilaté seulement.
- NBLØC : Nombre de bloc de 1024 valeurs du spectre a tracer par page.
- IFICH : Numéro du fichier sur la bande Benson. Le tracé sera compris entre les numéros d'étude 1 et (900 + IFICH).

REMARQUES :
. Le nombre de points de sortie, soit NPRIM=2^n est fixé dans le programme.
. Le tracé est mis sur la bande FT14. Sur cette bande ont peut accumuler plusieurs fichiers. Pour un fichier donné, le tracé commence toujours au bloc n° 1 et se termine au bloc n° (900 + IFICH).
. Si ICHEN1=0, il faut enlever les cartes contrôle (DD et SETUP) de la "TAPE".

TEST8 - pour enregistreur du "Type MAILLARD"



Pour la signification générale des paramètres, se reporter aux explications données pour le programme "TEST8" - CAS GENERAL -

PARTICULARITES DE CE PROGRAMME((*))

- Un nouveau test "I36" a été ajouté à ceux de la 1ère carte.

- Les cartes données 3 et 4 ne doivent exister que si I36=3. Elles signifient que :

à partir du bloc	1,	il faudra appliquer la gamme	6
"	"	3,	"	"	"	4
"	"	10,	"	"	"	3
"	"	15,	"	"	"	2

Si { I36 ≠ 3 : Nombre de 6 caractères y compris la gamme sur la "TAPE".
{ I36 = 3 : " 3 " non " " " " " "

REMARQUE : Si ICHEN1 = 0 (lecture de l'interf. sur la FT12) la valeur de I36 n'a aucune importance.

Le programme TEST8S correspondant utilise les quatre premières cartes données seulement.

P R O G R A M M E "BSS10"

- GENERALITES
- CARTES DE CONTROLE
- CARTES DONNEES

PROGRAMME "BSS10" - GENERALITES

Ce programme calcule tout l'intervalle spectral libre (1/2.h) correspondant à un interférogramme de "M" valeurs. L'intervalle spectral choisi est celui où se trouve la valeur "SIGB" à préciser dans les cartes données en cm^{-1}.

L'interférogramme peut être lu soit sur une bande "TAPE", soit sur une bande "FT12" suivant que le paramètre "ICHEN1" est égal à 1 ou à 0. Dans le cas où il est lu sur la "TAPE", il n'y a pas création d'une bande "FT12" comme dans le programme "TEST8".

Plusieurs programmes du type "BSS10" ont été prévus en fonction de la valeur de M désirée et de l'encombrement correspondant en mémoire centrale. Pour un programme "BSS10" donné, la valeur maximum de "M" est limitée à la puissance de 2 indiquée sur la tranche du paquet de cartes correspondant. On aura donc intérêt, une fois fixée la valeur de M, à choisir le programme dont la puissance de 2 indiquée se rapproche le plus de "M" tout en lui restant supérieure.

Dans tous les cas, le nombre de points de sortie du spectre calculé B sera égal à la puissance de 2 immédiatement supérieure ou égale à "M" (Valeur minimum possible : 1024).

Remarque :

On trouvera dans l'appendice D une note concernant les programmes distribués et l'estimation de leurs caractéristiques (encombrement, durée, nombre de lignes).

CARTES DE CONTROLE

BSS10 A PARTIR DE LA TAPE

```
//BSSE9 JOB COT0830,AMIOT,TIME=(,18),MSGLEVEL=(0,0)
/*SETUP UNIT=BDE7,DDNAME=TAPE,ID=(009238,NORING,SAVE,NL)
/*SETUP UNIT=BD16,DDNAME=FT14F001,ID=103182
                              {125K
// EXEC   FTHLG,REGION.GO=     {185K
                              {420K
//LKED.SYSIN DD *
     .....
      .....          {  4K
   'DECK' DU BSS10   { 16K
       .....         {  64K
        .....
/*
//GO.TAPE DD   UNIT=BDE7,DISP=(OLD,KEEP),DCB=(DEN=1,TRTCH=ET,BUFNO=1),
// LABEL=(1,NL)
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2)
//GO.IMP DD SYSOUT=A
//GO.SYSIN DD *
         10.10.71                                              10100
+0.60000000D+04       4000   13  907
+0.10000000D+01+0.30000000D+05   62    1    1
+0.10000000D+03+0.200E+02   907 KK R CH4+CO  VIDE  6K   4000  POINTS
/*
//
```

- Si les programmes sont en bibliothèque sur disque magnétique, voir en complément l'appendice G p. 59.

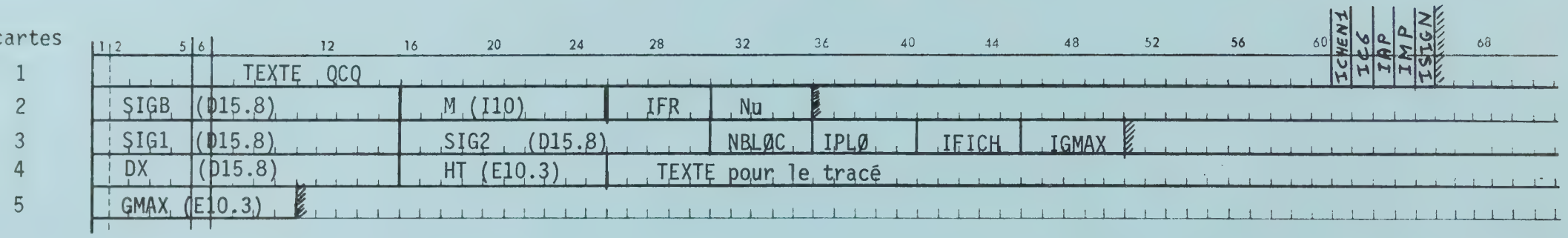
BSS10 A PARTIR DE LA FT12

```
//BSSE9 JOB CIR0780,VERGES,TIME=(,15),MSGLEVEL=(0,0)
/*SETUP UNIT=BD16,DDNAME=FT12F001,ID=(102587,NORING,SAVE,SL)
/*SETUP UNIT=BD16,DDNAME=FT14F001,ID=103182
                              (108K
// EXEC   FTHLG,REGION.GO=(168K
                              (404K
//LKED.SYSIN DD *
     .....
      .....          ( 4K
   'DECK' DU BSS10 ( 16K
       .....         ( 64K
        .....
/*
//GO.FT12F001 DD DSN=F123015,UNIT=BD16,DISP=OLD,LABEL=(09,SL),
//DCB=(BUFNO=2,RECFM=VS,LRECL=1004,BLKSIZE=1008)
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2)
//GO.IMP DD SYSOUT=A
//GO.SYSIN DD *
                                                          00100
+0.60000000D+04     16384     15000   11 3032
+0.10000000D+01+0.30000000D+05   62    1    1
+0.20000000D+02+0.150E+02 3032 23 06 71  15000 PTS PU239
/*
//
```

- Si les programmes sont en bibliothèque sur disque magnétique, voir en complément l'appendice G p. 59.

- Dans ce programme, on remplace les programmes de lecture de la "TAPE" (PLECB2, PLASS, FLAG par ex.) par un programme "BIDON" de 2 ou 3 cartes binaires. C'est ce qui permet de gagner de la place en mémoire par rapport au précédent.

BSS10 "CAS GENERAL"



. Voir la signification des paramètres dans le chapitre TEST8. p. 18.

SEULES DIFFERENCES :

IMP = { 1 sortie de toutes les valeurs de chaque bloc de 100 si ICHEN1 = 1.
{ 2 " des 10 premières valeurs " " " si ICHEN1 = 0.

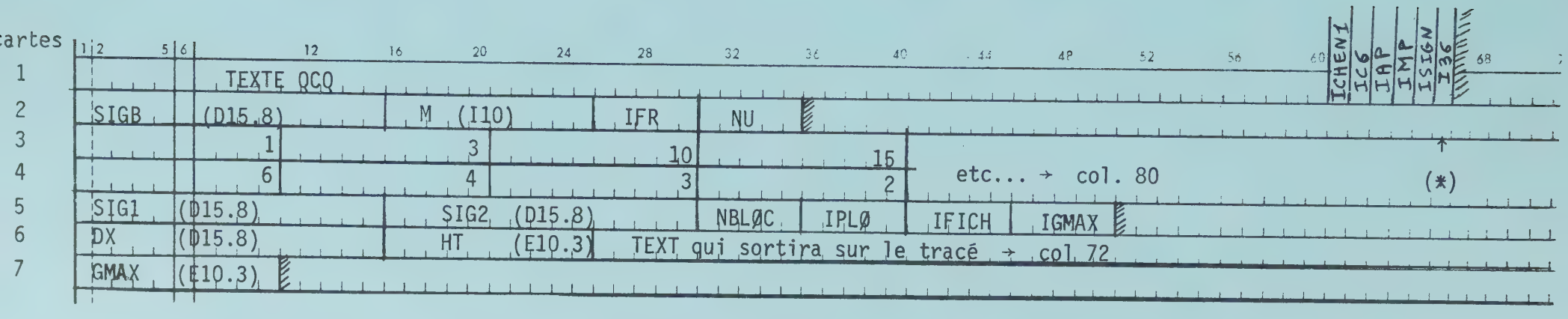
IGMAX = 0 la carte donnée numéro 5 n'est pas lue par le programme (et "GMAX" = 1).

IGMAX = 1 " " " " est lue par le programme. Toutes les valeurs du spectre seront divisées par "GMAX" avant d'être tracées.

IC6 = 1 valeurs primaires du spectre gardées sur bande FT16.

IC6 = 0 " " " non gardées sur bande FT16.

BSS10 - pour enregistreur "Type MAILLARD"



. Pour la signification des paramètres des cartes 1, 2, 5, 6, 7, se reporter aux explications données pour le programme "BSS10" - CAS GENERAL. p.24.

- Pour les cartes 3 et 4, qui ne sont à donner que si I36=3 : se reporter aux explications du programme "TEST8" Type MAILLARD, p. 19.

P R O G R A M M E "TFID3" et "T102"

- GENERALITES
- CARTES DE CONTROLE
- CARTES DONNEES

PROGRAMMES "TFID3" et "T102" - GENERALITES

Ces deux programmes séparés à l'origine ont été regroupés en un seul programme de 2 étapes appelé : "TFIDT102"

1ère ETAPE "T F I D 4"

Dans cette étape, on calcule la T.F. d'interférogrammes échantillonnés avec un grand nombre de valeurs. La valeur de M n'est théoriquement pas limitée, mais pour des raisons de capacité des disques magnétiques, la valeur maximum de M est : 2.10^6.

Les valeurs de l'interférogramme sont lues sur la bande magnétique "FT12". Lorsque le programme est terminé, on trouve les valeurs de la T.F. de l'interférogramme sur un disque magnétique. Elles sont réparties sur deux fichiers (FT01 et FT11) à l'état brut c'est-à-dire non normalisées à 1.

2ème ETAPE "T 1 0 2"

Ce programme exploite les NPRIM valeurs du spectre calculé par "TFD4". NPRIM est une puissance de 2 qui aura été calculée automatiquement par "TFID4" en fonction de M, mais qu'il faudra indiquer dans les cartes données de cette étape. NPRIM est tout simplement la puissance de 2 immédiatement supérieure ou égale à M.

Ce programme commence donc par lire les NPRIM/2 premières valeurs du spectre sur le fichier disque FT01 et les NPRIM/2 dernières sur le fichier disque FT11. Ayant trouvé la valeur maximum du spectre, il le normalise et écrit alors ses valeurs sur la bande magnétique FT16 par blocs de 1024 valeurs.

Il interpole les points primaires ainsi obtenus dans la région (SIG1) où il a trouvé la valeur maximum du spectre. Pour tracer les 1 ou 2 blocs de 1024 valeurs qui entourent SIG1, il calcule automatiquement l'échelle DX cm^{-1}/cm à prendre pour que le tracé ne dépasse pas un mètre de longueur. Le programme sort les deux valeurs SIC1 et DX, ainsi que le coefficient (GMAX) de réduction à appliquer aux valeurs interpolées pour qu'elles soient normalisées sur les tracés ultérieurs faits par le programme "MPLØT".

Il est possible de limiter la recherche de la valeur maximum du spectre à un domaine (SIGN1, SIGN2) précisé en CM^{-1} dans une carte donnée. Ceci afin d'éviter que le psectre soit normalisé sur une raie parasite ou un défaut prévu à l'avance grâce à l'observation du même spectre à basse résolution.

CARTES DE CONTROLE DU TFID-T102

```
//TFIDT102 JOB CIR1111,CHEVILLARD,TIME=(2,30)
/*SETUP DDNAME=FT01F001,UNIT=3330,ID=SET001
/*SETUP UNIT=BD16,DDNAME=FT32F001,ID=105552
/*SETUP  DDNAME=FT14F001,UNIT=BD16,ID=102026
//S1 EXEC FTHLG,REGION.GO=192K
//LKED.SYSLMOD DD DISP=(NEW,PASS)
//LKED.SYSIN DD *
     *****
      *****
   'DECK' DU TFID4.
      *****
       *****
/*
//GO.FT01F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF01T,VOL=SER=SET001
//GO.FT11F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF11T,VOL=SER=SET001
//GO.FT21F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF21T,VOL=SER=SET001
//GO.FT22F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(57,1)),
// DSNAME=CIR0719.DELOUIS.TF22T,VOL=SER=SET001
//GO.SYSIN DD *
    1    500000
/*

//S2 EXEC FTHLG,REGION.GO=192K
//LKED.SYSLMOD DD DISP=(OLD,PASS)
//LKED.SYSIN DD *
     *****
      *****
   'DECK' DU T102.
      *****
       *****
/*
//GO.FT01F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF01T,VOL=SER=SET001
//GO.FT11F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF11T,VOL=SER=SET001
//GO.FT16F001 DD UNIT=BD16,DISP=NEW,LABEL=(1,SL),DSN=FT16851D,
// VOL=SER=108275,
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2),LABEL=5
//GO.SYSIN DD *
    500000  512  851    1    5    1
   12+0.50000000D+041
+0.44000000D+04+0.54000000D+04
/*
//
```

au choix

```
//LKED.SYSLIN DD DSN=CIR0719.DELOUIS.TFID4,
// UNIT=3330,VOL=SER=SET001,DISP=OLD
//LKED.SYSLMOD DD DISP=(NEW,PASS)
```

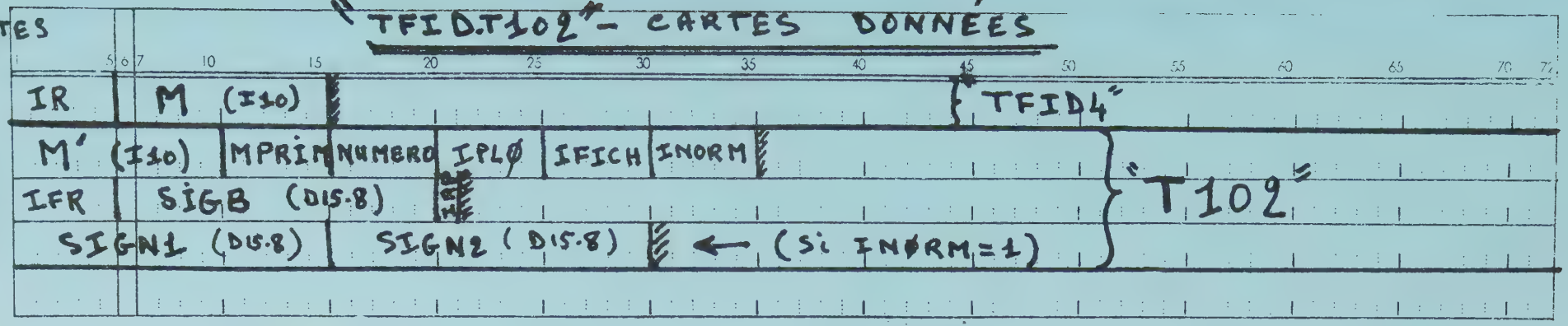
au choix

```
//LKED.SYSLIN DD DSN=CIR0719.DELOUIS.T102,
// UNIT=3330,VOL=SER=SET001,DISP=OLD
//LKED.SYSLMOD DD DISP=(OLD,PASS)
```

REMARQUES:- Les cartes /*SETUP et // DD de la bande Benson "FT14" sont à supprimer si on ne désire pas de tracé. (IPLØ=0 - Voir cartes données).

- En cas d'exécutions multiples, voir en complément l'appendice H page 67.

"TFID.T102" - CARTES DONNÉES



ère étape (carte n° 1) - PROGRAMME TFID4 -

Le programme "TFID4" ne sort qu'une des deux moitiés de l'intervalle spectral (I.S.) de largeur 1/h (où h est le pas de l'interf.).

- { IR=1 → 1ère moitié de l'I.S.
  IR=3 → 2ème moitié de l'I.S.
- M= nombre de valeurs de l'interf. à lire sur la bande "FT12" ou "FT32" avec 20 000 < M < 2097152

ème étape (cartes n° 2, 3, 4) - PROGRAMME T102 -

- MPRIM=NPRIM/1024 où NPRIM est la puissance de 2 ⩾ M qui est la valeur de la carte donnée n° 1.
- M'=M dans le cas général.
- M'<M si on veut diminuer la résolution (ne pas dépasser un fact. 2)
- NUMERO de l'Interf.
- IFR (voir notations générales).
- SIGB=abcisses qcq. de la tranche spectrale désirée.
- IPLØ=0 → pas de tracé → pas de "FT14" (/*SETUP et // DD à supprimer).
  IPLØ=1 → le tracé de la région où le spectre a été normalisé est fait automatiquement (longueur : environ 1 m). Prévoir la "FT14".
- IAP=0 pas d'apodisation et IAP=1 → apodisation de la région tracée par APØ1, [APØ2, APØ3 ou APØ4 (voir p.73).]
- INØRM≠1 → pour la normalisation, le maximum du spectre B est recherché entre B(500) et B(NPRIM-500). (Carte n° 4 ignorée).
- INØRM=1 → rajouter la carte n° 4 qui fixe en CM^{-1} les limites "SIGN1" et "SIGN2" entre lesquelles on cherchera la valeur maximum du spectre pour le normaliser.

2048 K	valeurs possibl de MPRIM
1024 K	
512 K	
256 K	
128 K	
64 K	
32 K	

URÉE A PREVOIR SUR 165/I.B.M. (le deuxième temps est celui de "T102").

{ MPRIM=2048 → 8'16" + 1'08"
  MPRIM=1024 → 4'04" + 36"

{ MPRIM=512 → 1'57" + 19"
  MPRIM=256 → 50" + 10"

{ MPRIM=128 → 24" + 6"
  MPRIM= 64 → 12" + 4"

PROGRAMME "MPLØT" ou "MPLØS"

- GENERALITES
- CARTES DE CONTROLE
- CARTES DONNEES

GENERALITES - PROGRAMME "MPLØT" ou MPLØS"

Ce programme permet de tracer un spectre une fois qu'il a été calculé par le programme "TFIDT101". Il utilise les points "primaires" du spectre enregistrés sur la bande "FT16". Les généralités sur les tracés décrits page 3 s'appliquent à ce programme.

Nous verrons à propos des cartes données de ce programme les détails concernant son exploitation et la mise en page du spectre.

PROGRAMME MPLØS - PARTICULARITES

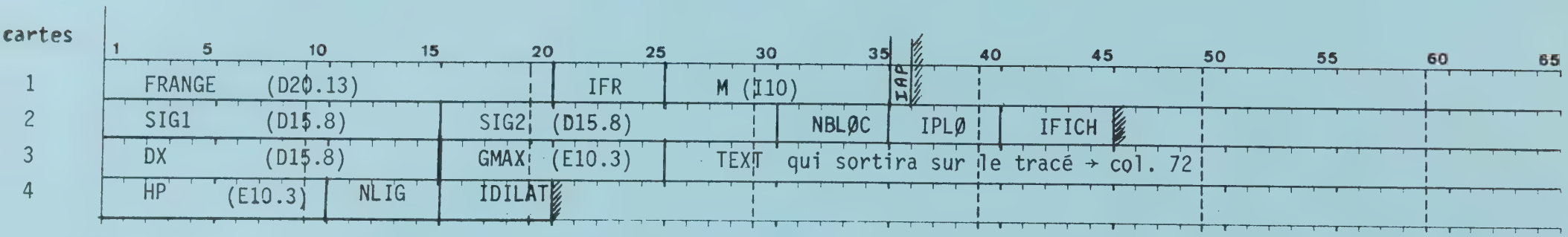
- Il trace de SIG1 à SIG2 exactement. (Voir cartes données et généralités page 3).
- NLIG=1 et NBLØC=99 valeurs imposées dans le programme.
- En fin de tracé, la plume revient au point de départ.
- Les valeurs négatives sont tronquées à - 20 cm.

CARTES DE CONTROLE

```
//MPLOT JOB CIF0320,DELOUIS,MSGLEVEL=(0,0)
/*SETUP UNIT=BD16,DDNAME=FT16F001,ID=(102547,NORING,SAVE,SL)
/*SETUP UNIT=BD16,DDNAME=FT14F001,ID=103182
// EXEC   FTHLG,REGION.GO=84K
//LKED.SYSIN DD *
       .....
        .....
    '' DECK ''DU PROGRAMME 'MPLOT' OU 'MPLOS'.
         .....
          .....
/*
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2)
//GO.FT16F001 DD UNIT=BD16,DISP=OLD,LABEL=(1,SL),DSN=FT16900,
// DCB=(BUFNO=2,RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.SYSIN DD *
+0.3507984664000D-05   60    8700001
+0.33640000D+04+0.33660000D+04    1    1    1
+0.50000000D-02+0.270E+01   920   HTE. RES.
+0.300E+02    1  100
/*
//
```

Si les programmes sont dans une bibliothèque sur disque, voir en complément l'appendice G p. 59.

MPLOT ou MPLOS



. Pour la signification générale des paramètres IFR, SIG1, SIG2, NBLØC, IAP, IPLØ, DX, TEXT, M se reporter aux paragraphes II et III, page 2 et 3 de cette brochure.

. Ce programme permet de tracer un spectre par "page" de plusieurs "lignes".

- Par "ligne" on entend : une courbe représentant une partie du spectre ((NBLØC * 1024) points primaires) avec son système d'axes et éventuellement la même courbe dilatée en superposition.

- Chaque "page" est constituée d'une ou plusieurs lignes décalées en hauteur.

- HP Hauteur totale du papier en CM.

- NLIG Nombre de lignes par page. (La hauteur de papier effectée à chaque ligne est calculée automatiquement à partir de HP et NLIG).

- IDILAT Si IPLØ = 9, le spectre sera tracé, multiplié par le facteur IDILAT.

- GMAX Prendre ici la valeur de "GMAX" listée par le programme "TFID.T101" qui a calculé ce spectre. Cette valeur permet de normaliser le spectre interpolé.

- M L'interf. correspondant au spectre sera pondéré sur M valeurs. Donc en principe prendre la valeur donnée pour le programme "TFID.T101". (Ou plus petit si on veut diminuer la résolution, mais sans dépasser un facteur 2).

- FRANGE L'unité de base des pas h de l'interf. est donnée une fois pour toute dans le programme T102. La valeur que l'on donne ici permet de la réajuster de façon à modifier les valeurs SIG1B, DELTAB (paragraphe IV, page 4) du spectre qui sont écrites sur la FT16. On peut ainsi réajuster les abscisses du spectre tracé.

COPIE DE FICHIERS BANDE

- Programme "UTICOPIE"
- Programme "B5COPIE"
- Programme "C6COPIE"

PROGRAMMES DE COPIES DE BANDES

PROGRAMME UTICOPIE

On peut effectuer la copie d'un fichier au moyen d'un utilitaire I.B.M. qui a pour nom IEBGENER. Le programme UTICOPIE emploie cet utilitaire. On peut avec UTICOPIE recopier un fichier d'une bande sur une autre bande. Ce programme est constitué uniquement de cartes de contrôle dans lesquelles il suffit d'indiquer les caractéristiques de la bande origine dans la carte DD "//SYSUT1" et de la bande copie dans la carte DD "//SYSUT2".

```
//UTICØPIE JØB CØTO830,GUELACH
/*SETUP UNIT=BD16,DDNAME=SYSUT1,ID=(1C3642,NCRING,SAVE,BLP)
/*SETUP UNIT=BC16,DDNAME=SYSUT2,ID=(1C3359,RING,SAVE,SL)
// EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DC DSN=FT16728,UNIT=BD16,LABEL=(9,BLP),VOL=SER=103642,
// DISP=(OLD,KEEP),DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032)
//SYSUT2 DD DSN=FT16728,UNIT=BD16,LABEL=(03,SL),VOL=SER=103359,
// DISP=(NEW,KEEP),DCB=(RECFM=VS,LRECL=1C28,BLKSIZE=1032)
//SYSUDUMP DD SYSCUT=A
/*
```

Ce programme est très court, il demande très peu de place en mémoire (environ 60 k).

Si la copie s'est bien passée, on doit avoir sur le listing les indications suivantes :

```
DATA SET UTILITY - GENERATE

IEB334I NO EDITING CR CCNVERSION WILL BE DONE
PRCCESSING ENDED AT ECD
```

PROGRAMMES FORTRAN

Contrairement à UTICOPIE, les 2 programmes FORTRAN de copie B5COPIE et C6COPIE s'appliquent exclusivement aux bandes FT12 et FT16.

Dans ces 2 programmes, il faut une carte SETUP par bande, mais une carte DD par fichier, puisqu'elle en définit les caractéristiques.

Pour les fichiers autres que le premier (pour chaque bande), il faut le paramètre VOL=SER=*.FT{16/22}F001 qui indique au système que les fichiers 2 et suivants sont sur la même bande que le premier. (Voir exemple de la page suivante à ce sujet).

PROGRAMME B5COPIE

Ce programme lit la bande origine et la recopie sur une autre. On peut par ce programme, recopier plusieurs fichiers consécutifs ou non. Le nombre de fichiers à recopier figure dans une carte donnée en format I2.

EXEMPLE DE COPIE DU 6ème fichier d'une B5 sur le FICHIER 1 d'une NOUVELLE BANDE

```
//B5COPIE JOB COT0330,GUELACH
/*SETUP UNIT=BD16,DDNAME=FT12F001,ID=(103391,NORING,SAVE,SL)
/*SETUP UNIT=BD16,DDNAME=FT22F001
// EXEC   FTHCLG,PARM.FORT='OPT=2',REGION.GO=78K
//FORT.SYSIN DD *
C
C      ************  PROGR. B5COPIE   ************
C    CE PROGRAMME PERMET DE RECOPIER LES KMAX FICHIERS D'UNE FT12.
C
      DIMENSION P(10000)
      READ 777 ,KMAX
  777 FORMAT(I2)
      PRINT 2,KMAX
    2 FORMAT(' ***** PROGR. B5COPIE -V. DU 29.08.72  ****',/,I4,' FICHIE
     1RS A RECOPIER')
      JPL=10000
   21 FORMAT('      BLOC',I6,' RECOPIE')
C
      DO 200 KI=1,KMAX
      READ(12)NU,M,NU,M,NU,M
      WRITE(22)NU,M,NU,M,NU,M
      PRINT 5 ,NU,M
    5 FORMAT(//,' NU=',I6,'  M=',I8,/)
      MK1=M/JPL
      MK2=M-MK1*JPL
      IF(MK1.EQ.0) GO TO 2001
      DO 1 L=1,MK1
      DO 30 I=1,JPL
   30 P(I)=0.
      READ(12,END=600)P
      WRITE(22)P
    1 PRINT 21,L
      IF(MK2.EQ.0) GO TO 300
 2001 DO 31 I=1,MK2
   31 P(I)=0.
      READ(12,END=700)(P(K),K=1,MK2)
      WRITE(22)(P(K),K=1,MK2)
      PRINT 4 ,MK2
    4 FORMAT(6X,'DERNIER BLOC DE',I6,' VAL. RECOPIE')
  300 READ(12,END=200)P
      GO TO 200
C
  600 PRINT 40,L
   40 FORMAT(' E.O.F. RENCONTREE EN VOULANT LIRE LE BLOC',I6)
      GO TO 200
  700 PRINT 41,MK2
   41 FORMAT(' E.O.F. RENCONTREE EN VOULANT LIRE LE BLOC DE',I6,'VAL.')
C
  200 END FILE 22
      PRINT 6
    6 FORMAT(/,' FIN DE LA COPIE')
      STOP
      END
/*
//GO.FT12F001 DD UNIT=BD16,DISP=OLD,LABEL=(06,SL),DSN=FT12862,
// DCB=(BUFNO=2,RECFM=VS,LRECL=1004,BLKSIZE=1008)
//GO.FT22F001 DD UNIT=BD16,DISP=NEW,LABEL=(01,SL),DSN=FT12862,
// DCB=(BUFNO=2,RECFM=VS,LRECL=1004,BLKSIZE=1008)
//GO.SYSUDUMP DD SYSOUT=A
//GO.SYSIN DD *
01
/*
```

PROGRAMME C6COPIE

Ce programme est similaire à B5COPIE mais copie exclusivement des C6 (FT16).

EXEMPLE DE COPIE D'UNE C6 SUR UNE AUTRE BANDE : Dans cet exemple, on recopie les 6 fichiers de la 102024 sur la 104436 après les 2 fichiers qui s'y trouvent déjà.

```
//C6COPIE JOB COT0830,GUELACH,TIME=(,30),MSGLEVEL=(0,0)
/*SETUP UNIT=BD16,DDNAME=FT16F001,ID=(102024,NORING,SAVE,SL)
/*SETUP UNIT=BD16,DDNAME=FT22F001,ID=104436
// EXEC FTHCLG,REGION.GO=70K
//FORT.SYSIN DD *
      DIMENSION P(1024)
      REAL*8 SIG1B,DELTAB
      READ 777,KMAX
  777 FORMAT(I2)
      KI=0
  100 READ(16)NU,NU,NU,ND1024,SIG1B,DELTAB
      WRITE(22)NU,NU,NU,ND1024,SIG1B,DELTAB
       PRINT 77,NU,ND1024,SIG1B,DELTAB
   77 FORMAT(' NU=',I8,' ND1024=',I9,' SIG1B=',D15.8,' DELTAB=',D15.8)
      KI=KI+1
       ND1025=ND1024+1
       DO 1 L=1,ND1025
      N=16
      READ(N,END=200)P
   20 WRITE(22)P
      PRINT 2,L
    2 FORMAT('      BLOC',I7,' RECOPIE')
    1 CONTINUE
  200 END FILE 22
      IF(KI.NE.KMAX) GO TO 100
      PRINT 78
   78 FORMAT(10X,'FIN DE LA COPIE')
      STOP
      END
/*
//GO.FT16F001 DD UNIT=BD16,DISP=(OLD,KEEP),DSN=FT16MAY1,LABEL=(1,SL)
//GO.FT16F002 DD UNIT=BD16,DISP=(OLD,KEEP),DSN=FT16MAY2,LABEL=(2,SL),
// VOL=REF=*.FT16F001
//GO.FT16F003 DD UNIT=BD16,DISP=(OLD,KEEP),DSN=FT16MAY3,LABEL=(3,SL),
// VOL=REF=*.FT16F001
//GO.FT16F004 DD UNIT=BD16,DISP=(OLD,KEEP),DSN=FT16MAY4,LABEL=(4,SL),
// VOL=REF=*.FT16F001
//GO.FT16F005 DD UNIT=BD16,DISP=(OLD,KEEP),DSN=FT16MAY5,LABEL=(5,SL),
// VOL=REF=*.FT16F001
//GO.FT16F006 DD UNIT=BD16,DISP=(OLD,KEEP),DSN=FT16MAY6,LABEL=(6,SL),
// VOL=REF=*.FT16F001
//*
//GO.FT22F001 DD UNIT=BD16,DISP=(NEW,KEEP),DSN=FT16MAY1,LABEL=(3,SL),
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.FT22F002 DD UNIT=BD16,DISP=(NEW,KEEP),DSN=FT16MAY2,LABEL=(4,SL),
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032),VOL=REF=*.FT22F001
//GO.FT22F003 DD UNIT=BD16,DISP=(NEW,KEEP),DSN=FT16MAY3,LABEL=(5,SL),
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032),VOL=REF=*.FT22F001
//GO.FT22F004 DD UNIT=BD16,DISP=(NEW,KEEP),DSN=FT16MAY4,LABEL=(6,SL),
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032),VOL=REF=*.FT22F001
//GO.FT22F005 DD UNIT=BD16,DISP=(NEW,KEEP),DSN=FT16MAY5,LABEL=(7,SL),
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032),VOL=REF=*.FT22F001
//GO.FT22F006 DD UNIT=BD16,DISP=(NEW,KEEP),DSN=FT16MAY6,LABEL=(8,SL),
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032),VOL=REF=*.FT22F001
//GO.SYSIN  DD *
06
```

A P P E N D I C E A

- Création d'une bande au C.I.R.C.E.
- Demande d'un tracé au dispatching
- Note remise au dispatching quand les opérateurs ont été obligés d'arrêter un JOB ("CANCEL-OPERATOR")

CREATION D'UNE BANDE AU C.I.R.C.E.

Lorsque vous aurez besoin d'une nouvelle bande (ce qui sera souvent le cas au début), vous devrez faire ce qu'on appelle une "création". Ceci consiste à :

1°) remplir une étiquette (au dispatching) comme suit :

UTILISATEUR	SIGLE	NUMÉRO CALCUL
GUELACH	COT	830
LABEL	ENREGISTREMENT 7 canaux / 9 canaux	DENSITÉ
(SL) NL BLP		200 - 500 - 800 (1600)

LABEL

SL.

Cette étiquette doit être fixée sur la carte rose par un coin. Ne pas la coller entièrement car elle doit être mise sur la bande créée.

Les cases non remplies varient suivant les caractéristiques de la bande (pour une FT12 ou une FT16, c'est binaire, 1600 ; pour une FT14, c'est DCB, 800).

2°) mettre dans la carte SETUP correspondante, à la place du numéro, l'indidation SCRATC (ou SCRATH ou n'importe quelle configuration de 6 caractères alphabétiques commençant par les 3 lettres SCR) à la place du numéro de la bande. (On peut aussi supprimer le paramètre ID).

Exemple :

CARTE SETUP pour une bande dont on connaît déjà le numéro :

```
/*SETUP  UNIT-BD16,DDNAME=FT12F001,ID=(103275,SAVE,RING,SL)
```

CARTE SETUP pour une bande qu'on veut créer :

```
/*SETUP  UNIT=BD16,DDNAME=FT12F001,ID=(SCRATC,SAVE,RING,SL)
ou  /*SETUP  UNIT=BD16,DDNAME=FT12F001
```

Au retour du paquet dans le casier au dispatching, on doit trouver le numéro de la bande créée dans la colonne "fichiers à créer" qui aura été préalablement remplie comme suit :

Nom de l'utilisateur : GUELACH | Nom du travail : TEST8 | Durée du calcul : .. H .. M 20 S

370/165 - CENTRE INTER REGIONAL DE CALCUL ELECTRONIQUE - 370/165

UTILISATEUR	SIGLE	NUMÉRO CALCUL
GUELACH	COT	830
LABEL	ENREGISTREMENT 7 canaux / 9 canaux	DENSITÉ
(SL) NL BLP		200 - 500 - 800 (1600)

LABEL

SL

Fichiers à créer sur bande

(joindre une étiquette par bande)

UNITE (BDE9/BDE7/BD16)	LABEL (SL/NL/BLP)	VOLUME SERIAL
BD16	SL	

DEMANDE D'UN TRACE AU DISPATCHING :

TRACE BENSON
****

NOM DE L'UTILISATEUR	SIGLE DU LABORATOIRE	NUMERO DE CALCUL	DATE ET HEURE DE LA DEMANDE

NUMERO DE LA BANDE =

DENSITE=(DEN=2 (800 BPI) ~~OU DEN=1 (500 BPI)~~). ((1))

NOMBRE (N-1)= DE FICHIERS A SAUTER SUR LA BANDE. ((2))

NUMERO DU BLOC PERMETTANT DE REPERER LA FIN D'UN FICHIER = 900 +IFICH

NUMERO DU BLOC CORRESPONDANT AU DEBUT DU TRACE = 1

NUMERO DU DERNIER BLOC A TRACER = 900

LARGEUR DU PAPIER (30CM OU 70CM). ((1))

PAPIER (BLANC DU MILLIMETRE). ((1))

GROSSEUR DE LA PLUME EN MM (2/10,3/10,4/10,6/10, OU 8/10) ((1))

ENCRE (ROUGE,BLEU,VERTE, OU NOIRE). ((1))

DUREE APPROXIMATIVE DU TRACE EN MINUTES = nombre d'E/S divisé par 10

LONGUEUR APPROXIMATIVE DU TRACE EN METRES= (Voir "listing")

REMARQUES PARTICULIERES
****

((1))...RAYER LES MENTIONS INUTILES.
((2))...POUR N REGARDER DANS LA CARTE DD LE PARAMETRE LABEL=(N,SL).

PARTIE A REMPLIR PAR L'OPERATEUR
****

DATE ET HEURE DE RESTITUTION=

DUREE DU TRACE =

LONGUEUR (1M,5M,10M,20M,PLUS DE 20M). ((1))

* Note remise par les opérateurs (dans une boîte placée sur le guichet du dispatching) lorsqu'ils ont arrêté un job en cours d'exécution ("CANCEL OPERATOR").

.I.R.C.E

DATE LE 14/09/197

NOM DU JOB GLIEJUMP-1
1 2

N. DU JOB 3

VOTRE TRAVAIL A ETE COUPE

AU NIVEAU	*	POUR LE MOTIF
	*	
	*BANDE	9247 DATA CHECK.
~~SETUP~~	*	
	*~~DISQUE~~	
(EXPLOITATION)	*	
	*	
	*	
~~LISTING~~	*	
	*	

L OPERATEUR

A P P E N D I C E B

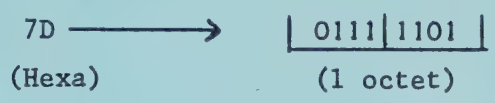
- DUMP par "DUMPBDE"
- Décodage d'un DUMP HEXA
- Analyse des "Sense Bytes"

} "TAPE" 7 pistes

DUMP D'UNE BANDE PAR "DUMPBDE"

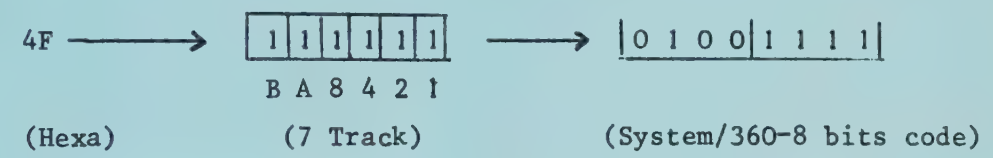
Lors de la lecture d'une bande, son image est transférée en mémoire centrale dans des groupes de 8 bits appelés "octets". Sur la liste d'un "DUMP", chaque octet peut être représenté par deux caractères hexadécimaux. Les pages 38 et 39 suivantes donnent la correspondance entre ces deux représentations.

Exemple :



Dans le cas de la lecture d'une bande 7 pistes, une correspondance arbitraire est établie entre les configurations à 6 bits de cette bande (7-Track Tape BCDIC) et les configurations à 8 bits des octets en mémoires centrale. Cette correspondance se trouve aussi dans les pages 38 et suivantes.

Exemple :



En cas d'erreur détectée, le programme "DUMPBDE" donne la liste en hexadécimal d'un certain nombre d'octets de contrôle appelés "Sense Bytes" où se trouvent codés certains renseignements concernant les conditions dans lesquelles et pour lesquelles s'est produite l'erreur. Le code de ces "Sense Bytes" se trouve dans les pages 38 à 43 suivantes. Un exemple montre comment il faut les utiliser.

DECODAGE D'UN DUMP HEXA

RR FORMAT INSTRUCTIONS

Decimal	Hexadecimal	Mnemonic	Graphic & Control Symbols BCDIC	Graphic & Control Symbols EBCDIC	(2) 7-Track Tape BCDIC	Punched Card Code	System/360 8-Bit Code
0	00			NUL		12-0-1-8-9	0000 0000
1	01			SOH		12-1-9	0000 0001
2	02			STX		12-2-9	0000 0010
3	03			ETX		12-3-9	0000 0011
4	04	SPM		PF		12-4-9	0000 0100
5	05	BALR		HT		12-5-9	0000 0101
6	06	BCTR		LC		12-6-9	0000 0110
7	07	BCR		DEL		12-7-9	0000 0111
8	08	SSK				12-8-9	0000 1000
9	09	ISK				12-1-8-9	0000 1001
10	0A	SVC		SMM		12-2-8-9	0000 1010
11	0B			VT		12-3-8-9	0000 1011
12	0C	(EBCDIC +)		FF		12-4-8-9	0000 1100
13	0D	(EBCDIC -)		CR		12-5-8-9	0000 1101
14	0E			SO		12-6-8-9	0000 1110
15	0F			SI		12-7-8-9	0000 1111
16	10	LPR		DLE		12-11-1-8-9	0001 0000
17	11	LNR		DC1		11-1-9	0001 0001
18	12	LTR		DC2		11-2-9	0001 0010
19	13	LCR		TM		11-3-9	0001 0011
20	14	NR		RES		11-4-9	0001 0100
21	15	CLR		NL		11-5-9	0001 0101
22	16	OR		BS		11-6-9	0001 0110
23	17	XR		IL		11-7-9	0001 0111
24	18	LR		CAN		11-8-9	0001 1000
25	19	CR		EM		11-1-8-9	0001 1001
26	1A	AR		CC		11-2-8-9	0001 1010
27	1B	SR		CU1		11-3-8-9	0001 1011
28	1C	MR		IFS		11-4-8-9	0001 1100
29	1D	DR		IGS		11-5-8-9	0001 1101
30	1E	ALR		IRS		11-6-8-9	0001 1110
31	1F	SLR		IUS		11-7-8-9	0001 1111
32	20	LPDR		DS		11-0-1-8-9	0010 0000
33	21	LNDR		SOS		0-1-9	0010 0001
34	22	LTDR		FS		0-2-9	0010 0010
35	23	LCDR				0-3-9	0010 0011
36	24	HDR		BYP		0-4-9	0010 0100
37	25	LRDR		LF		0-5-9	0010 0101
38	26	MXR		ETB		0-6-9	0010 0110
39	27	MXDR		ESC		0-7-9	0010 0111
40	28	LDR				0-8-9	0010 1000
41	29	CDR				0-1-8-9	0010 1001
42	2A	ADR		SM		0-2-8-9	0010 1010
43	2B	SDR		CU2		0-3-8-9	0010 1011
44	2C	MDR				0-4-8-9	0010 1100
45	2D	DDR		ENQ		0-5-8-9	0010 1101
46	2E	AWR		ACK		0-6-8-9	0010 1110
47	2F	SWR		BEL		0-7-8-9	0010 1111
48	30	LPER				12-11-0-1-8-9	0011 0000
49	31	LNER				1-9	0011 0001
50	32	LTER		SYN		2-9	0011 0010
51	33	LCER				3-9	0011 0011
52	34	HER		PN		4-9	0011 0100
53	35	LRER		RS		5-9	0011 0101
54	36	AXR		UC		6-9	0011 0110
55	37	SXR		EOT		7-9	0011 0111
56	38	LER				8-9	0011 1000
57	39	CER				1-8-9	0011 1001
58	3A	AER				2-8-9	0011 1010
59	3B	SER		CU3		3-8-9	0011 1011
60	3C	MER		DC4		4-8-9	0011 1100
61	3D	DER		NAK		5-8-9	0011 1101
62	3E	AUR				6-8-9	0011 1110
63	3F	SUR		SUB		7-8-9	0011 1111

(2) Add C (check bit) for odd or even parity as needed, except for even parity, decimal 64 is CA, the same as decimal 122.
(3) Decimal Feature instructions. (4) System 360 assembler programs require these codes.

RR FORMAT

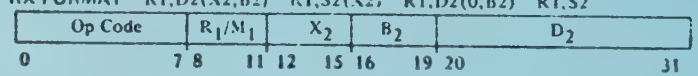
R1, R2 – meaningful for all RR instructions except SPM, SVC

BASE AND INDEX REGISTERS

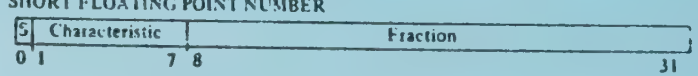
RX FORMAT INSTRUCTIONS

Decimal	Hexadecimal	Mnemonic	Graphic & Control Symbols BCDIC	Graphic & Control Symbols EBCDIC	(2) 7 Track Tape BCDIC	Punched Card Code	System/360 8-bit Code	(4)
64	40	STH		SP	(2)	no punches	0100 0000	
65	41	LA				12-0-1-9	0100 0001	
66	42	STC				12-0-2-9	0100 0010	
67	43	IC				12-0-3-9	0100 0011	
68	44	EX				12-0-4-9	0100 0100	
69	45	BAL				12-0-5-9	0100 0101	
70	46	BCT				12-0-6-9	0100 0110	
71	47	BC				12-0-7-9	0100 0111	
72	48	LH				12-0-8-9	0100 1000	
73	49	CH				12-1-8	0100 1001	
74	4A	AH	¢	¢		12-2-8	0100 1010	
75	4B	SH	.	.	B A 8 2 1	12-3-8	0100 1011	
76	4C	MH	<	<	B A 8 4	12-4-8	0100 1100	
77	4D		(	(	B A 8 4 1	12-5-8	0100 1101	(
78	4E	CVD	+	+	B A 8 4 2	12-6-8	0100 1110	+
79	4F	CVB	|	!	B A 8 4 2 1	12-7-8	0100 1111	
80	50	ST	&	&	B A	12	0101 0000	
81	51					12-11-1-9	0101 0001	
82	52					12-11-2-9	0101 0010	
83	53					12-11-3-9	0101 0011	
84	54	N				12-11-4-9	0101 0100	
85	55	CL				12-11-5-9	0101 0101	
86	56	O				12-11-6-9	0101 0110	
87	57	X				12-11-7-9	0101 0111	
88	58	L				12-11-8-9	0101 1000	
89	59	C				11-1-8	0101 1001	
90	5A	A	!	!		11-2-8	0101 1010	
91	5B	S	$	$	B 8 2 1	11-3-8	0101 1011	
92	5C	M	*	*	B 8 4	11-4-8	0101 1100	
93	5D	D	)	)	B 8 4 1	11-5-8	0101 1101	)
94	5E	AL	;	;	B 8 4 2	11-6-8	0101 1110	
95	5F	SL	Δ	¬	B 8 4 2 1	11-7-8	0101 1111	
96	60	STD	-	-	B	11	0110 0000	
97	61		/	/	A 1	0-1	0110 0001	
98	62					11-0-2-9	0110 0010	
99	63					11-0-3-9	0110 0011	
100	64					11-0-4-9	0110 0100	
101	65					11-0-5-9	0110 0101	
102	66					11-0-6-9	0110 0110	
103	67	MXD				11-0-7-9	0110 0111	
104	68	LD				11-0-8-9	0110 1000	
105	69	CD				0-1-8	0110 1001	
106	6A	AD				12-11	0110 1010	
107	6B	SD	,	,	A 8 2 1	0-3-8	0110 1011	
108	6C	MD	%	%	A 8 4	0-4-8	0110 1100	
109	6D	DD	¥	_	A 8 4 1	0-5-8	0110 1101	
110	6E	AW	\	>	A 8 4 2	0-6-8	0110 1110	
111	6F	SW	¤	?	A 8 4 2 1	0-7-8	0110 1111	
112	70	STE				12-11-0	0111 0000	
113	71					12-11-0-1-9	0111 0001	
114	72					12-11-0-2-9	0111 0010	
115	73					12-11-0-3-9	0111 0011	
116	74					12-11-0-4-9	0111 0100	
117	75					12-11-0-5-9	0111 0101	
118	76					12-11-0-6-9	0111 0110	
119	77					12-11-0-7-9	0111 0111	
120	78	LF				12-11-0-8-9	0111 1000	
121	79	CE				1-8	0111 1001	
122	7A	AE	`	:	A	2-8	0111 1010	
123	7B	SE	#	#	8 2 1	3-8	0111 1011	
124	7C	ME	@	@	8 4	4-8	0111 1100	
125	7D	DE	'	'	8 4 1	5-8	0111 1101	'
126	7E	AU	=	=	8 4 2	6-8	0111 1110	=
127	7F	SU	"	"	8 4 2 1	7-8	0111 1111	

RX FORMAT R1,D2(X2,B2) R1,S2(X2) R1,D2(0,B2) R1,S2

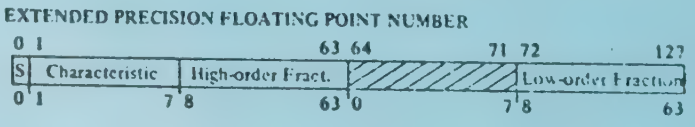


SHORT FLOATING POINT NUMBER



LONG FLOATING POINT NUMBER – same as short floating point number except fraction is longer – bits 8 through 63

EXTENDED PRECISION FLOATING POINT NUMBER



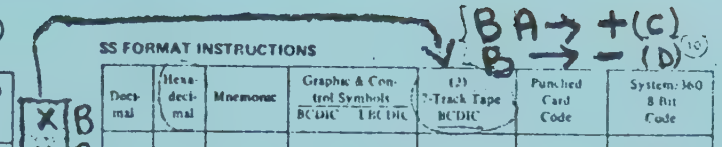
DECODAGE D'UN DUMP HEXA

RS, SI FORMAT INSTRUCTIONS

Decimal	Hexadecimal	Mnemonic	Graphic & Control Symbols BCDIC	Graphic & Control Symbols EBCDIC	(2) 7-Track Tape BCDIC	Punched Card Code	System/360 8-Bit Code
128	80	SSM				12-0-1-8	1000 0000
129	81		a			12-0-1	1000 0001
130	82	LPSW	b			12-0-2	1000 0010
131	83	(Diagnose)	c			12-0-3	1000 0011
132	84	WRD	d			12-0-4	1000 0100
133	85	RDD	e			12-0-5	1000 0101
134	86	BXH	f			12-0-6	1000 0110
135	87	BXLE	g			12-0-7	1000 0111
136	88	SRL	h			12-0-8	1000 1000
137	89	SLL	i			12 0 9	1000 1001
138	8A	SRA				12-0-2-8	1000 1010
139	8B	SLA				12-0-3-8	1000 1011
140	8C	SRDL				12-0-4-8	1000 1100
141	8D	SLDL				12-0-5-8	1000 1101
142	8E	SRDA				12-0-6-8	1000 1110
143	8F	SLDA				12-0-7-8	1000 1111
144	90	STM				12-11-1-8	1001 0000
145	91	TM	j			12-11-1	1001 0001
146	92	MVI	k			12-11-2	1001 0010
147	93	TS	l			12-11-3	1001 0011
148	94	NI	m			12 11-4	1001 0100
149	95	CLI	n			12-11-5	1001 0101
150	96	OI	o			12-11-6	1001 0110
151	97	XI	p			12-11-7	1001 0111
152	98	LM	q			12-11-8	1001 1000
153	99		r			12-11-9	1001 1001
154	9A					12-11-2-8	1001 1010
155	9B					12-11-3-8	1001 1011
156	9C	SIO				12-11-4-8	1001 1100
157	9D	TIO				12-11-5-8	1001 1101
158	9E	HIO				12-11-6-8	1001 1110
159	9F	TCH				12-11-7-8	1001 1111
160	A0					11-0-1-8	1010 0000
161	A1					11-0-1	1010 0001
162	A2		s			11-0-2	1010 0010
163	A3		t			11-0-3	1010 0011
164	A4		u			11-0-4	1010 0100
165	A5		v			11-0-5	1010 0101
166	A6		w			11-0-6	1010 0110
167	A7		x			11-0-7	1010 0111
168	A8		y			11-0-8	1010 1000
169	A9		z			11-0-9	1010 1001
170	AA					11-0-2-8	1010 1010
171	AB					11-0-3-8	1010 1011
172	AC					11-0-4-8	1010 1100
173	AD					11-0-5-8	1010 1101
174	AE					11-0-6-8	1010 1110
175	AF					11-0-7-8	1010 1111
176	B0					12-11-0-1-8	1011 0000
177	B1					12-11-0-1	1011 0001
178	B2					12-11-0-2	1011 0010
179	B3					12-11-0-3	1011 0011
180	B4					12-11-0-4	1011 0100
181	B5					12-11-0-5	1011 0101
182	B6					12-11-0-6	1011 0110
183	B7					12-11-0-7	1011 0111
184	B8					12-11-0-8	1011 1000
185	B9					12-11-0-9	1011 1001
186	BA					12-11-0-2-8	1011 1010
187	BB					12-11-0-3-8	1011 1011
188	BC					12-11-0-4-8	1011 1100
189	BD					12-11-0-5-8	1011 1101
190	BE					12-11-0-6-8	1011 1110
191	BF					12-11-0-7-8	1011 1111

SS FORMAT INSTRUCTIONS

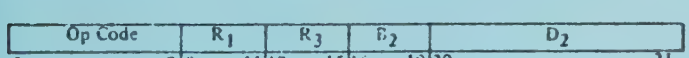
Decimal	Hexadecimal	Mnemonic	Graphic & Control Symbols BCDIC	Graphic & Control Symbols EBCDIC	(2) 7-Track Tape BCDIC	Punched Card Code	System/360 8 Bit Code
192	C0		?		BA8 2	12-0	1100 0000
193	C1		A	A	BA 1	12-1	1100 0001
194	C2		B	B	BA 2	12-2	1100 0010
195	C3		C	C	BA 21	12-3	1100 0011
196	C4		D	D	BA 4	12-4	1100 0100
197	C5		E	E	BA 4 1	12-5	1100 0101
198	C6		F	F	BA 42	12-6	1100 0110
199	C7		G	G	BA 421	12-7	1100 0111
200	C8		H	H	BA8	12 8	1100 1000
201	C9		I	I	BA8 1	12-9	1100 1001
202	CA					12-0-2-8-9	1100 1010
203	CB					12-0-3-8-9	1100 1011
204	CC					12-0-4-8-9	1100 1100
205	CD					12-0-5-8-9	1100 1101
206	CE					12-0-6-8-9	1100 1110
207	CF					12-0-7-8-9	1100 1111
208	D0		!		B 8 2	11-0	1101 0000
209	D1	MVN	J	J	B 1	11-1	1101 0001
210	D2	MVC	K	K	B 2	11-2	1101 0010
211	D3	MVZ	L	L	B 21	11-3	1101 0011
212	D4	NC	M	M	B 4	11-4	1101 0100
213	D5	CLC	N	N	B 4 1	11-5	1101 0101
214	D6	OC	O	O	B 42	11-6	1101 0110
215	D7	XC	P	P	B 421	11-7	1101 0111
216	D8		Q	Q	B 8	11 8	1101 1000
217	D9		R	R	B 8 1	11-9	1101 1001
218	DA					12-11-2-8-9	1101 1010
219	DB					12-11-3-8-9	1101 1011
220	DC	TR				12-11-4-8-9	1101 1100
221	DD	TRT				12-11-5-8-9	1101 1101
222	DE	ED (3)				12-11-6-8-9	1101 1110
223	DF	EDMK (3)				12-11-7-8-9	1101 1111
224	E0		‡		A8 2	0-2-8	1110 0000
225	E1					11-0-1-9	1110 0001
226	E2		S	S	A 2	0-2	1110 0010
227	E3		T	T	A 21	0-3	1110 0011
228	E4		U	U	A 4	0-4	1110 0100
229	E5		V	V	A 4 1	0-5	1110 0101
230	E6		W	W	A 42	0-6	1110 0110
231	E7		X	X	A 421	0-7	1110 0111
232	E8		Y	Y	A8	0-8	1110 1000
233	E9		Z	Z	A8 1	0-9	1110 1001
234	EA					11-0-2-8-9	1110 1010
235	EB					11-0-3-8-9	1110 1011
236	EC					11-0-4-8-9	1110 1100
237	ED					11-0-5-8-9	1110 1101
238	EE					11-0-6-8-9	1110 1110
239	EF					11-0-7-8-9	1110 1111
240	F0		0	0	8 2	0	1111 0000
241	F1	MVO	1	1	1	1	1111 0001
242	F2	PACK	2	2	2	2	1111 0010
243	F3	UNPK	3	3	21	3	1111 0011
244	F4		4	4	4	4	1111 0100
245	F5		5	5	4 1	5	1111 0101
246	F6		6	6	42	6	1111 0110
247	F7		7	7	421	7	1111 0111
248	F8	ZAP (3)	8	8	8	8	1111 1000
249	F9	CP (3)	9	9	8 1	9	1111 1001
250	FA	AP (3)				12-11-0-2-8-9	1111 1010
251	FB	SP (3)				12-11-0-3-8-9	1111 1011
252	FC	MP (3)				12-11-0-4-8-9	1111 1100
253	FD	DP (3)				12-11-0-5-8-9	1111 1101
254	FE					12-11-0-6-8-9	1111 1110
255	FF					12-11-0-7-8-9	1111 1111



RS FORMAT

R1,R3,D2(B2) } BXH, BXLE
R1,R3,S2 } LM, STM

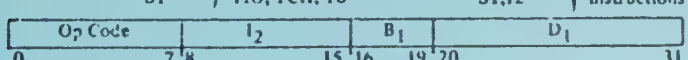
R1,D2(B2) } Shift instructions
R1,S2 }



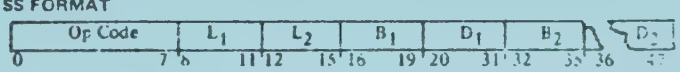
SI FORMAT

D1(B1) } LFSW, SSM, HIO, SIO
S1 } TIO, TCH, TS

D1(B1),I2 } All other SI instructions
S1,I2 }



SS FORMAT



D1(L,B1),D2(B2) } NC, OC, XC, CLC, MVC, MVN, MVZ, TR, TRT, ED, EDMK
S1(L),S2

D1(L1,B1),D2(L2,B2) } PACK, UNPK, MVO, AP, CP, DP, MP, SP, ZAP
S1(L1),S2(L2)

PACKED DECIMAL NUMBER — digit | digit ---- digit | digit | digit | sign

ZONED DECIMAL NUMBER — zone | digit ---- zone | digit | sign | digit

ANALYSE DES "SENSE BYTES" → (erreur dans un DUMP)

Sense Byte 0

Bit	Designation	2401-2404, 2415 and 2420 Interpretation	
		Models 1-3 (and Models 4-6, 800 bpi, NRZI Mode)	Models 4-7 (1,600 bpi PE)
0	Command reject	Set when a write, write tape mark, or erase command is addressed to a file protected tape unit, or a data-converter-on control command that is addressed to a seven-track tape unit is recognized on a TCU with the seven-track compatibility feature but without the data converter feature. In this case, mode set is executed for parity, density, and translator.	Same as Models 1-3
1	Intervention required	Set whenever tape unit status A is inactive, i.e., tape unit is not ready or nonexistent. See "Sense Byte 1."	Same as Models 1-3
2	Bus-out check	Set whenever even parity appears on the information bus lines from the channel to the control unit.	Same as Models 1-3
3	Equipment check (excluding 2415) Equipment check (2415)	Set when reject tape unit (bit 1, byte 4) or sequence error (bit 5, 6, or 7 of byte 4) is set. Set whenever C-compare or clock check occurs. See "Sense Byte 3."	Set when reject tape unit (bit 1, byte 4) is set. Same as Models 1-3
4	Data check	Set when a data check occurs. See "Sense Byte 3."	Same as Models 1-3; see "Sense Byte 4."
5	Overrun	Set if service is requested, but data cannot be transferred during a read, write, or read backward operation. Data transfer stops as soon as condition is detected. Note: Data check during overrun suppresses the overrun indication.	Same as Models 1-3
6	Word count zero	Set during a write operation if transfer of data is prevented before the first byte of data. When word count zero is set, no tape motion occurs.	Same as Models 1-3
7	Data converter check	See "Data Conversion Feature."	Same as Models 1-3

Sense Byte 1

Bit	Designation	2401-2404, 2415 and 2420 Interpretation	
		Models 1-3 (and Models 4-6, 800 bpi, NRZI Mode)	Models 4-7 (1,600 bpi, PE)
0	Noise (excluding 2415)	During a read forward space block, indicates that data was recognized after the normal LRC byte time but not long enough after to be considered a new block. Data before the LRC byte is checked and transferred; data after the LRC byte turns on the noise bit and maintains tape motion, but is not transferred. When connected to Model 2 control, during a read backward or backspace block, if data is recognized after the disconnect sequence is started. With Model 1 control, data recognized after start of disconnect, is transferred as part of block. Noise bit is not set; data check is probable.	Set during read or read backward if a data check occurs.
		During a write, erase gap, or write tape mark, indicates that data (or noise caused by tape defects) was detected at the read head before the block or tape mark was written, or during erase gap while the tape was being erased. Data check and unit check are indicated.	Same as Models 1-3
0	Noise (2415)	During a write or write tape mark, indicates that data (or noise caused by tape defects) was detected at the read head before the block or tape mark was written.	Same as Models 1-3
1*	TU Status A	Selected and ready.	Same as Models 1-3
2*	TU Status B (non-2415)	Not ready, or rewinding, or under the control of another TCU via the 2816 Switching Unit. Assuming no outstanding device end status, the bits determine response to initial selection as follows:	Same as Models 1-3

1* et 2* regroupés ensembles TSVP.

BDE9 - BDE7 - TAPE(CS) → dérouleur 2415 II } Type des lecteurs

BD16 (FT16 ou FT12) → dérouleur 3420 (7) } de bande

"SENSE BYTE" en HEXA :

08 → | 0 0 0 0 | 1 0 0 0 | (voir exemple) } Numérotation des bits

bit→ 0 1 2 3 4 5 6 7

ANALYSE DES "SENSE BYTES" (suite)

2*	TU Status B	Tape Unit Status A / Tape Unit Status B / Tape Unit Status / Response to Initial Selections 0 / 0 / Nonexistent / Unit check 0 / 1 / Not ready / Unit check, arm for device end 1 / 0 / Ready and not rewinding and not switched. * / Clear status 1 / 1 / Ready and rewinding or switched or power is down on a tape unit attached through a switching unit. / Busy, arm for device end Note: Unit check is not signaled for a sense operation. Following unit check or busy indication, device end will be signaled when the tape unit becomes ready and not rewinding. (2415) No references to the 2816 Switching Unit apply.	
3	Seven-track	The selected tape unit has the seven-track feature installed.	Same for 2415 Always 0 for 2401-2402 Models 4-7.
4	Load point	The selected tape unit is at load point.	Same as Models 1-3
5	Selected and write status	The selected tape unit is in write status.	Same as Models 1-3
6	File protect	The selected tape unit is in file protect status.	Same as Models 1-3
7	Not capable	Not used, always set to zero.	Tape unit and/or control are not compatible with mounted tape (indicated after read from load point).

* Switched means that the tape unit is selected by some other control unit under control of a tape unit switch.

Sense Byte 2

2401-2404, 2415 and 2420 Interpretation	
Models 1-3 (and Models 4-6, 800 bpi, NRZI Mode)	Models 4-7 (1,600 bpi, PE)
This sense byte contains the track-in-error indicator bits that are set at the end of a read or read backward command if a data check has been encountered. See "Cyclic Redundancy Check." At the end of a properly executed read or read backward with no data check and at the end of all other commands, sense byte 2 contains at least bits 6 and 7 set to 1's. No error correction is attempted when operating with seven-track tape units; bits 6 and 7 are set to 1's in sense byte 2.	Not applicable
2415: Not applicable; bits 6 and 7 set to 1 unconditionally.	

Sense Byte 3

		2401-2404, 2415 and 2420 Interpretation	
Bit	Designation	Models 1-3 (and Models 4-6, 800 bpi, NRZI Mode)	Models 4-7 (1,600 bpi, PE)
0	R/W VRC (2401-4 Models 1-6) Data reg VRC (2415 Models 1-6)	A vertical redundancy check occurred during a read or read backward operation. Indicator is not set after an overrun or after receipt of a stop signal. *	A vertical parity error occurred during read or read backward operation that could not be corrected, or a Model 7 write operation.
1	LRCR (Models 1-3) Multiple Track Error (Models 4-6)	A longitudinal redundancy check occurred during write, write tape mark, read, or read backward operation.	Weak signal in more than one track on a read or read backward operation. Data is incorrect. Also velocity check on write operation.

ANALYSE DES "SENSE BYTES" (suite)

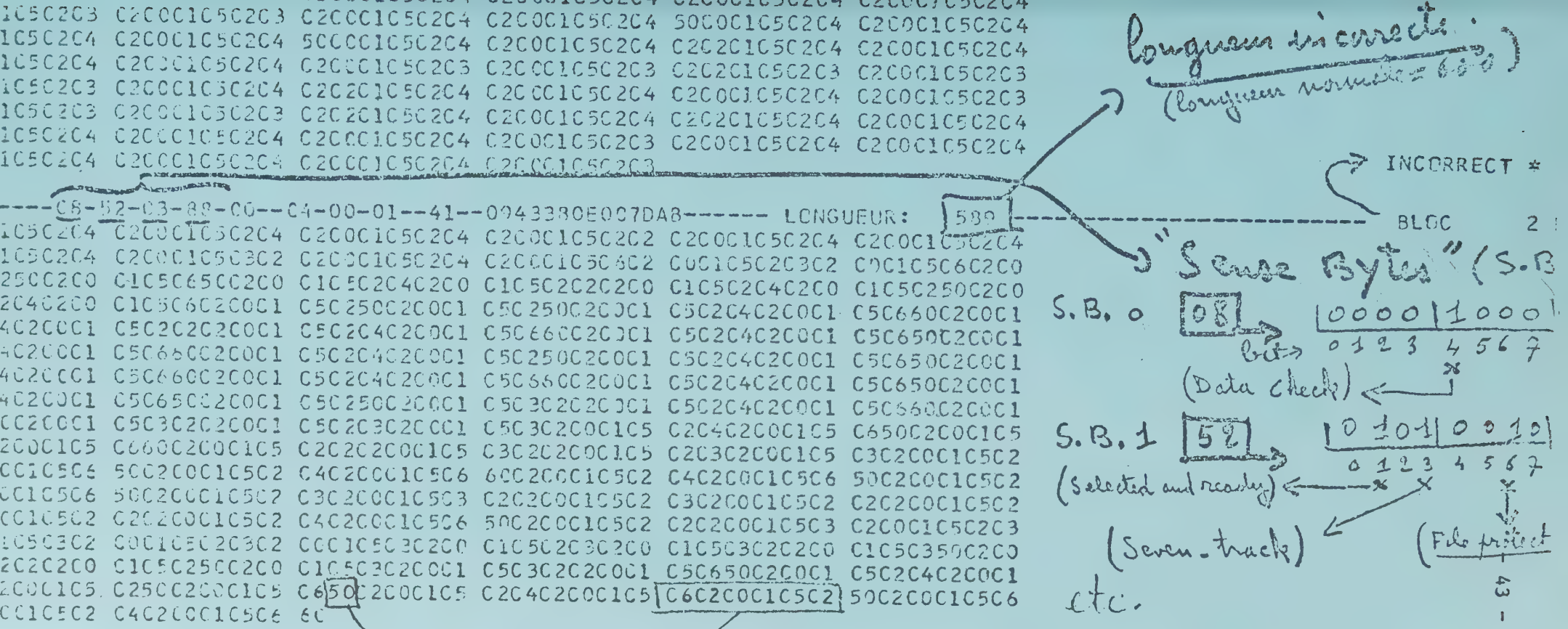
2	Skew	Excessive skew detected by a read back check during a write, write tape mark, or erase operation.	Excessive skew detected during a read or read backward operation.
3	CRC (Models 1-3) End data check (Models 4-6)	A cyclic redundancy check occurred during a read or read backward operation (nine-track only). 2415: Not applicable.	Set when sync burst following data block is not properly recognized, or is improperly recognized before actual end of data, 2415: Set with false end-of-block indication.
4	Skew reg VRC (Models 1-3) Envelope check (Models 4-6)	A character with incorrect parity detected in skew register during write, write tape mark, or erase operation.	Indicates at lease one track with low signal while writing.
4 (2415)	Read reg VRC (Models 1-3) Envelope check (Models 4-6)	A character with incorrect parity detected in read register during write or write tape mark operation.	Indicates at least one track with low signal while writing.
5	Phase encoding	Not applicable; always set to zero.	Selected tape unit is set to PE recording mode.
6	Backward	The selected tape unit is in backward status.	Same as Models 1-3
7	C compare	C compare is a data check. It indicates that parity of data into the register did not equal that out of the data register.	Same as Models 1-3
Note: Bits 0-4 and 7 of byte 3 indicate data checks. Any of these will set data check (bit 4, byte 0). 2415: Bit 7 causes equipment check.			
* Stop signal resulted from halt I/O instruction in CPU.			

Sense Byte 4

		2401-2404, 2415 and 2420 Interpretation	
		Models 1-3	Models 4-7
Bit	Designation	(and Models 4-6, 800 bpi, NRZI Mode)	(1,600 bpi, PE)
0	Not used	Same	Same as Models 1-3
1	Reject TU	Selected tape unit failed to respond to set read or set write status when instructed, or became not ready during execution of a tape motion operation. Equipment check (bit 3, byte 0) also set.	Same as Models 1-3
2	Read clock	Maintenance aids	Not used
3	Write clock		Maintenance aid
4	Delay counter		Not used
5	C sequence	These sequence indicators are maintenance aids. They are set only in the event of a machine failure and cause equipment check and unit check. 2415: All zero bits transmitted for sense byte 4.	Set in event of a machine failure and cause a data check and unit check.
6	B sequence		
7	A sequence		

Sense Byte 5

		2401-2404, 2415 and 2420 Interpretation	
		Models 1-3	Models 4-7
Bit	Designation	(and Models 4-6, 800 bpi, NRZI Mode)	(1,600 bpi, PE)
0	Reserved for future use	Always zero	Always zero
1	Reserved for CE function		
2-6	None	Always zero	Always zero
7	Reserved for RPQ use	Always zero	Always zero



EXEMPLE D'ANALYSE DES "SENSE BYTES" en cas d'erreurs sur une bande magnétique.

Convention de signe

BA → + (C en hexa)

B → - (D en hexa)

Le signe est répété avec chacun des 6 caractères du nombre.

C6 → 1 octet | 1 1 0 0 (C) | 0 1 1 0 (6) | → code 7 pistes

correct

1	B
1	A
0	8
1	4
1	2
0	1

50 → 1 octet | 0 1 0 1 (5) | 0 0 0 0 (0) |

erreur car tout caractère hexa, doit par convention commencer par C. ou D. Ici seul le signe a été envoyé sur la bande.

1
1
0
0
0
0

A P P E N D I C E C

- TAPE DUMP (TD)

(extrait du manuel de référence IBM H20-0321-6)

TAPE DUMP (TD)

The Tape Dump support program permits the dumping of a seven-track or a nine-track tape in either EBCDIC or hexadecimal character sets. TD performs in a conversational mode with the capability of tape positioning before dumping. The format of the message to initiate the Tape Dump DSP is:

*CALL,TD[,keyword-parameters]

The message text may include any combination of the following keyword parameters (if the parameter is omitted, the underlined value is assumed):

[IN={device-name|device-address|24007|<u>24009</u>}]

Input tape unit. Device name or device address may be specified as defined by the Support Processor. The use of 24007 or 24009 causes allocation of the first available seven-track or nine track tape drive.

[OUT={device-name|device-address|<u>1403</u>}]

Printer to be used. This parameter may be specified as 1403 (first available printer) or by device name or device address as defined by the Support Processor.

[ID=xxxxxx]

External label identification of tape to be dumped. If this parameter is omitted, mounting instructions will not clearly identify the tape to be mounted.

[DEN={2|5|<u>8</u>|16}]

Tape density:

2	200 bpi (seven-track only)
5	556 bpi (seven-track only)
8	800 bpi
16	1600 bpi (nine-track only)

[MD={C|T|C|E|ET}]

Tape recording technique:

C	Odd parity, converter on, translator off
T	Odd parity, converter off, translator on
O	Odd parity, converter off, translator off →(Tapes 9 canaux)
E	Even parity, converter off, translator off
ET	Even parity, converter off, translator on →(Tapes 7 canaux)

This parameter is not applicable to nine-track tapes.

A positioning parameter request is initially issued by TD after it has been called. The format of the request is:

JOB nnnn, REQUEST AT CTP F=nn, R=nnnnnn

where:

JOB = job number

REQUEST = request for tape positioning

CTP = current tape position

F = current file position (the first file = 01)

R = current record position within the current file

Initially, TD rewinds the tape to load point. In addition, TD operates only on single reel, multiple file tapes.

Processing is initiated and positioning parameters are supplied by issuing this command:

*START TD[,positioning-parameters]

The message text may include any combination of the following positioning parameters:

[F=±nn]

Number of files to be skipped. If plus is indicated, the tape is positioned forward nn files. If minus is specified, the tape is backed up nn files. In the absence of a sign, plus is assumed. If the parameter is omitted, the tape remains positioned at the current file.

[R=±nnnnnn]

Number of records to be skipped within the current file. Plus indicates forward positioning (n) records from the current tape position. Minus indicates backward positioning (n) records from the current position. In the absence of a sign, plus is assumed. If file positioning has been requested, the tape will be positioned at record number 1. Otherwise, the tape remains at the current record position.

[N=nnnnnn]

The number of records to be dumped after the tape is positioned. If this parameter is omitted, the tape is dumped from the current file position to end-of-file.

[M={H|E}]

The mode of the dump. H specifies hexadecimal character sets, and E specifies EBCDIC character sets.

[LL={132|n}]

Print line length desired, from 1 to 132 characters in EBCDIC mode. If hexadecimal mode is requested, the LL parameter must be expressed as half of the desired print line length. That is, LL=66 represents 132 printed characters or 66 hexadecimal characters. Thus the default value of LL for hexadecimal dumps is 66.

Positioning parameters may violate file or physical tape boundaries. Refer to Appendix B for messages relating to such violations.

The Tape Dump print format is:

- Page header line:

  JOB nnnn,TAPE DUMP,TAPE ID=xxxxxx,PAGE=nnnnnn

- Block header line:

  ------FILE NO nn, RECORD NO nnnnnn------

- Data

- Block trailer line (number of characters in block):

  nnnnnn CHARACTERS

- Special print lines. When applicable, the following are printed:

  *** EOF ***

  *** EOF *** END OF DATA OR REQUEST ISSUED AT EOF

  *** EOT ***

  *** PARITY ***

When the Tape Dump program is scheduled (via the call), it prints a mount message indicating the tape unit and printer to be used. When both units have been readied, the operator issues this command:

*START,TD[,positioning-parameters]

Program execution may be terminated at any time, except during a pending intervention on the tape unit, by issuing:

*CANCEL,TD or *RESTART,TD

In addition, dumping of the current file may be suspended by issuing:

*START,TD,FREE

If dump suspension is requested, TD suspends the dump and issues a new positioning request reflecting the position, in the current file, at which the dump was halted.

TD is dynamic in the sense that once a positioning request has been satisfied, dumping begins immediately. That is, positioning is not a standalone function but is linked to dumping.

The following sequence illustrates the use of the Tape Dump program:

1. To initialize Tape Dump for a seven-track, 556 bpi tape, even parity, converter off, and translator on, issue this message:

   *CALL,TD,ID=123456,DEN=5,MD=ET,IN=24007

2. TD will issue a mount message for the allocated tape unit and printer:

   JOB (job-number),JOB (job-name),INPUT ID=123456(180), OUTPUT(01E)

A P P E N D I C E D

Note à propos des programmes TEST8 et BSS10

NOTE A PROPOS DES PROGRAMMES TEST8 et BSS10

Pour simplifier l'écriture des cartes de contrôle, les paquets de cartes concernant ces deux programmes ont été prévus pour des exploitations partant ou bien de la TAPE ou bien de la FT12.

Les premiers sont marqués : "A PARTIR DE LA TAPE"
Les seconds sont marqués : "A PARTIR DE LA FT12"

Dans les programmes "A PARTIR DE LA FT12", les sous-programmes de lecture de la "TAPE" ont été enlevés ce qui explique leur taille réduite.

Dans le cas du "TEST8", la lecture de l'interf. sur la "FT12" est imposée par le programme, et aucune erreur n'est possible.

Dans le cas du "BSS10", il faut néanmoins mettre à 1 le test "ICHENI" (voir cartes données) sous peine d'erreur.

LISTE DES PROGRAMMES DISTRIBUES (non limitative)

BSS10 4 K - à partir de la TAPE
BSS10 4 K - à partir de la FT12
BSS10 16 K - à partir de la FT12
32 K - Type maillard (PP seulement)
TEST8 4 K - à partir de la TAPE
TEST8 4 K - à partir de la FT12
TEST8 16 K - à partir de la TAPE
TEST8 16 K - à partir de la FT12

TEST8S

DUMPBDE, UTICOPIE, B5COPIE, C6COPIE

MPLØT, MPLØS

TFID102

Pour les passages au CIRCE, voir appendice G page 59.

QUELQUES EXEMPLES DE DUREE DES PROGRAMMES SUR I.B.M. 370/165

1°) TEST8 - 16K avec M=10^6 à partir de la "FT12"
durée : 24 s région GØ = 192 K 100 lignes

2°) TEST8S - Pour décoder et recopier 10.000 valeurs,
il faut prévoir $\{^{0,88\ sec}_{0,30\ sec}\}$ pour une "TAPE" $\{^{7\ pistes}_{9\ pistes}\}$. (Région GØ ≃ 86 K)

3°) TEST8 - 4 K avec M=10^6 à partir de la "TAPE" 7 pistes.
durée : 1 mn 50 s région GØ = 140 K 106 lignes

4°) BSS10 - 4 K avec M=3800 à partir de la "TAPE" 9 pistes.
durée : 3 s région GØ = 115 K 106 lignes

5°) BSS10 - 16 K avec M=16.000 à partir de la "TAPE" 9 pistes.
durée : 8 s région GØ = 160 K 107 lignes si IMP=0

APPENDICE E

Extrait du "programmer's guide" à propos des fichiers fortran

• Table 15. Specifications Made by the FORTRAN Programmer for Record Types and Blocking

Step	Blocked or Unblocked	Record Type	RECFM Specification	Record Length	Buffer Length
Compiler or Linkage Editor	Unblocked	Fixed-Length	not specified	not specified[1]	BLKSIZE=record length
	Blocked	Fixed-Length	not specified[1]	not specified[1]	BLKSIZE=xxxx
Load Module Execution	Unblocked	Fixed-Length	RECFM=F[2]	BLKSIZE=xxxx[2]	BLKSIZE=xxxx
		Variable-Length	RECFM=V	LRECL=xxxx	
		Variable-Length Spanned	RECFM=VS	LRECL=xxxx	
		Undefined	RECFM=U	BLKSIZE=xxxx	
	Blocked	Fixed-Length	RECFM=FB	LRECL=xxxx	
		Variable-Length	RECFM=VB		
		Variable-Length Spanned	RECFM=VSB		
		Undefined	Blocked undefined records are not permitted		

[1]Permanently specified by the compiler and cannot be altered (see "DCB Assumptions for Load Module Execution").
[2]Not specified for direct-access data sets.

For blocked variable-length or fixed-length records used by load modules, the record length is specified by

LRECL=xxxx

block length and buffer length are specified by

BLKSIZE=xxxx

Undefined records cannot be blocked.

Table 15 is a summary of the specifications made by the programmer for record types and blocking in FORTRAN processing.

FORTRAN Records and Logical Records

In FORTRAN, records for sequential data sets are defined by specifications in FORMAT statements and by READ/WRITE lists. A record defined by a specification in a FORMAT statement is a FORTRAN record (see the FORTRAN IV Language publication, Order No. GC28-6515). A record defined by a READ/WRITE list is a logical record. Within each category, there are three types of records: fixed-length, variable-length, and undefined. In addition, fixed-length and variable-length records can be blocked.

For unformatted READ and WRITE statements the logical record, as defined by the I/O list, is placed into physical records and, if required, the logical record is spanned over physical records. When spanning occurs, FORTRAN library routines do not split-write an item over the span even though there is enough room in the buffer to accomodate part of the item. However, FORTRAN does provide the ability to read items split across segments.

FORMAT CONTROL

The following discussion provides information on records written under control of a FORMAT statement.

UNBLOCKED RECORDS: For fixed-length and undefined records, the record length and buffer length are specified in the BLKSIZE subparameter. For variable-length records, the record length is specified in the LRECL subparameter; the buffer length in the BLKSIZE subparameter. The information coded in a FORMAT statement indicates the FORTRAN record length (in bytes).

Fixed-Length Records: For unblocked fixed-length records written under FORMAT control, the FORTRAN record length must not exceed BLKSIZE (see Figure 32).

Example: Assume BLKSIZE=44

```
10  FORMAT(F10.5,I6,2F12.5,'SUMS')
    WRITE(20,10)AB,NA,AC,AD
```

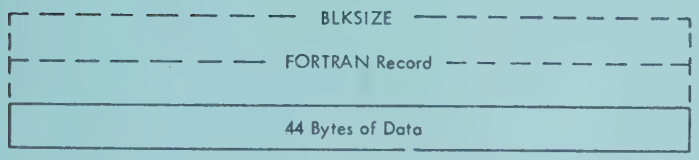


Figure 32. FORTRAN Record (FORMAT Control) Fixed-Length Specification

If the FORTRAN record length is less than BLKSIZE, the record is padded with blanks to fill the remainder of the buffer (see Figure 33). The entire buffer is written.

Example: Assume BLKSIZE=56

```
5  FORMAT(F10.5,I6,F12.5,'TOTAL')
   WRITE(15,5)BC,NB,BD
```

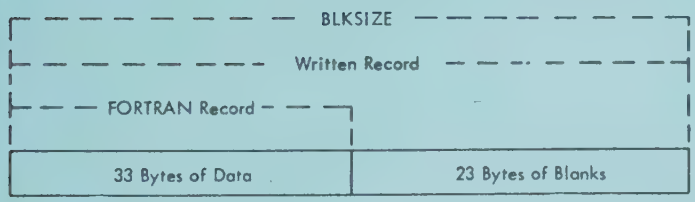


Figure 33. FORTRAN Record (FORMAT Control) Fixed-Length Specification and FORTRAN Record Length Less Than BLKSIZE

Variable-Length Records: For unblocked variable-length records written under FORMAT control, LRECL is specified as four greater than the maximum FORTRAN record length; and BLKSIZE as four greater than LRECL. These extra eight bytes are required for the 4-byte block descriptor word (BDW) and the 4-byte segment descriptor word (SDW), as shown in Figure 34. The BDW (see Figure 39) contains the length of the block; the SDW (see Figure 40) contains the length of the record segment, i.e., the data length plus four bytes for the SDW.

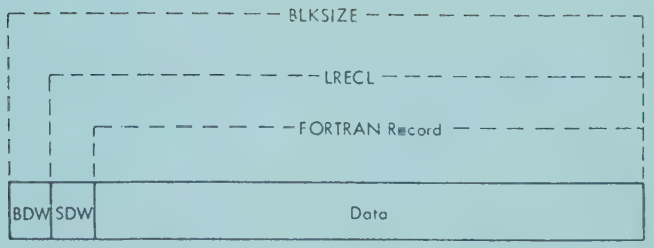


Figure 34. FORTRAN Record (FORMAT Control) Variable-Length Specification

If the data length is less than (LRECL-4), the unused portion of the buffer is not written (see Figure 35).

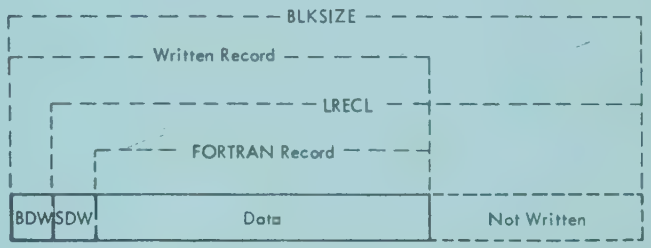


Figure 35. FORTRAN Record (FORMAT Control) With Variable-Length Specification and the FORTRAN Record Length Less Than (LRECL-4)

Undefined Records: For undefined records written under FORMAT control, BLKSIZE is specified as the maximum FORTRAN record length. If the FORTRAN record length is less than BLKSIZE, the unused portion of the buffer is not written (see Figure 36).

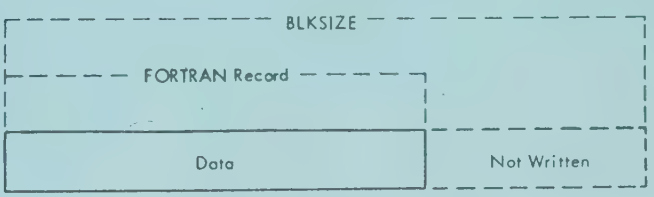


Figure 36. FORTRAN Record (FORMAT Control) With Undefined Specification and the FORTRAN Record Length Less Than BLKSIZE

BLOCKED RECORDS: For all blocked records, the record length is specified in the LRECL subparameter; the block length and buffer length in the BLKSIZE subparameter.

Fixed-Length Records: For blocked fixed-length records written under FORMAT control, LRECL is specified as maximum possible FORTRAN record length, and BLKSIZE must be an integral multiple of LRECL. If the FORTRAN record length is less than LRECL, the rightmost portion of the record is padded with blanks (see Figure 37).

Example: Assume BLKSIZE=48 and LRECL=24

```
10  FORMAT(I2,F4.1,F8.4,F10.5)
20  FORMAT(I3,F9.4)
      .
      .
      .
    WRITE(13,10)N,B,Q,S
      .
      .
      .
    WRITE(13,20)K,Z
```

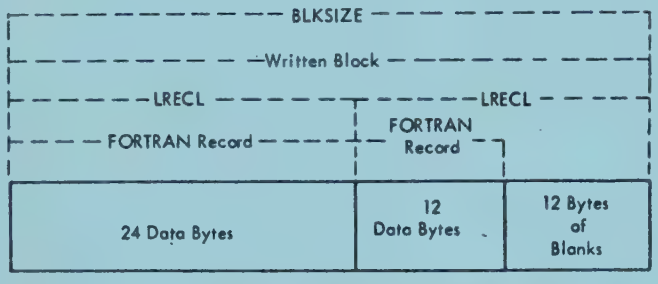


Figure 37. Fixed-Length Blocked Records Written Under FORMAT Control

Variable-Length Records: For blocked variable-length records written under FORMAT control, LRECL is specified as four greater than the maximum FORTRAN record length, and BLKSIZE must be 4 plus an integral multiple of LRECL. The four additional bytes allocated with BLKSIZE are required for the block descriptor word (BDW) that contains the block length. The four additional bytes allocated with LRECL are used for the segment descriptor word (SDW) that contains the record length indication.

If a WRITE is executed and the amount of space remaining in the present buffer is less than LRECL, only the filled portion of this buffer is written (see Figure 38); the new data goes into the next buffer. However, if the space remaining in a buffer is greater than LRECL, the buffer is not written, but held for the next WRITE (see Figure 38). If another WRITE is not executed before the job step is terminated, then the filled portion of the buffer is written.

If LRECL is omitted, its default value is set almost equal to the value of BLKSIZE. This results in having only one record written in any block.

Example: Assume BLKSIZE=28 and LRECL=12

```
30  FORMAT(I3,F5.2)
40  FORMAT(F4.1)
50  FORMAT(F7.3)
    WRITE(12,30)M,Z
    WRITE(12,40)V
    WRITE(12,50)Y
```

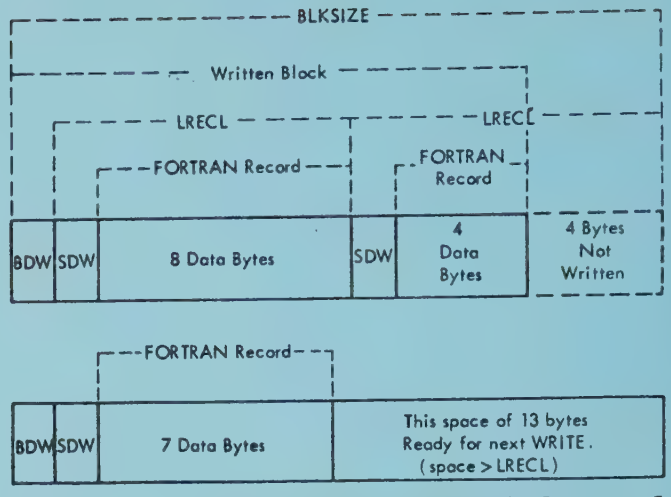


Figure 38. Variable-Length Blocked Records Written Under FORMAT Control

UNFORMATTED CONTROL

Only variable-length records can be written without format control, i.e., the RECFM subparameter must be VS. (If nothing is specified, VS is assumed.)

Records written with no FORMAT control have the following properties:

- The length of the logical record is controlled by the type and number of variables in the input/output list of its associated READ or WRITE statement.
- A logical record can be physically recorded on an external medium as one or more record segments. Not all segments of a logical record must fit into the same physical record (block).
- Two quantities control the manner in which records are placed on an external medium: the block size (as specified by the BLKSIZE parameter), and the logical record (as defined by the length of the I/O list). BLKSIZE is specified as part of the DCB parameter of the data definition (DD) statement. If not specified, FORTRAN provides default values.

Each block begins with a 4-byte block descriptor word (BDW); each segment begins with a 4-byte segment descriptor word (SDW). The SDWs and BDWs are provided by the system. Each buffer begins with a 4-byte block descriptor word (BDW). The SDWs and BDWs are provided by the system.

The format of a BDW is given in Figure 39.

Figure 39. Format of a Block Descriptor Word (BDW)

where:

block-length
is a binary count of the total number of bytes of information in the block. This includes four bytes for the BDW plus the sum of the segment lengths specified in each SDW in the block. (The permissible range is from 8 to 32,760 bytes.)

reserved
is two bytes of zeros reserved for system use.

The format of an SDW is given in Figure 40.

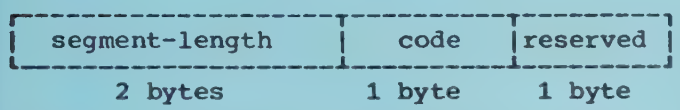


Figure 40. Format of a Segment Descriptor Word (SDW)

where:

segment-length
is a binary count of the number of bytes in the SDW (four bytes) plus the number of bytes in the data portion of the segment following the SDW. (The permissible range is from 4 to 32,756 bytes.)

code
indicates the position of the segment with respect to the other segments (if any) of the record. Bits 0 through 5 are reserved for system use and are set to 0. Bits 6 and 7 contain the codes:

Code	Meaning
00	This segment is not followed or preceded by another segment of the record.
01	This segment is the first of a multisegment record.
10	This segment is the last of a multisegment record.
11	This segment is neither the first nor last of a multi-segment record.

reserved
is a byte of zeros reserved for system use.

Unblocked Records: For unblocked records, if the logical record length is less than or equal to the length of the block (allowing four bytes for the BDW and four bytes for the SDW), the block will contain the entire logical record. The remainder of the block is unused and is not transmitted. The next record is placed in the following block in the same manner (see Figure 40.1).

If the logical record length is greater than the length of the block, the record is divided into record segments. The number of segments is determined from the READ/WRITE statement and the BLKSIZE specification. Again, only the filled portion of the block is transmitted and the next record is placed in the following block (see Figure 40.2).

Example: Assume BLKSIZE=40

```
REAL*4 A,B,C,D,E,F,G,H
  .
  .
  .
WRITE(9) A,B,C,D
  .
  .
  .
WRITE(9) E,F,G,H
```

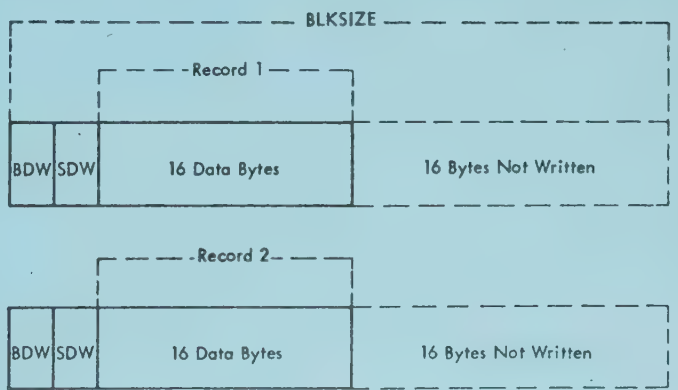
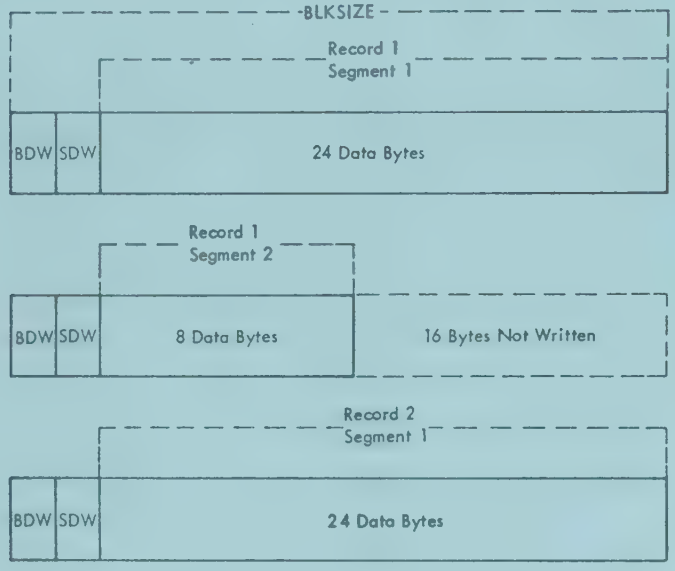


Figure 40.1. Unblocked Records Written Without FORMAT Control

Example: Assume BLKSIZE=32

```
REAL*8 A,B,C,D,E,F,G,H
  .
  .
  .
WRITE(9) A,B,C,D
  .
  .
  .
WRITE(9) E,F,G,H
```

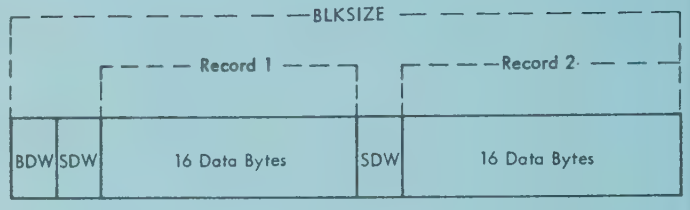


• Figure 40.2. Unblocked Segmented Records Written Without FORMAT Control

Blocked Records: For blocked records, if the logical record length is less than or equal to the length of the block (allowing four bytes for the BDW and four bytes for the SDW), the block will contain the entire logical record. The next record, preceded by its SDW, begins in the same block (see Figure 40.3).

Example: Assume BLKSIZE=44

```
REAL*8 A,B,C,D
  .
  .
  .
WRITE(9) A,B
  .
  .
  .
WRITE(9) C,D
```

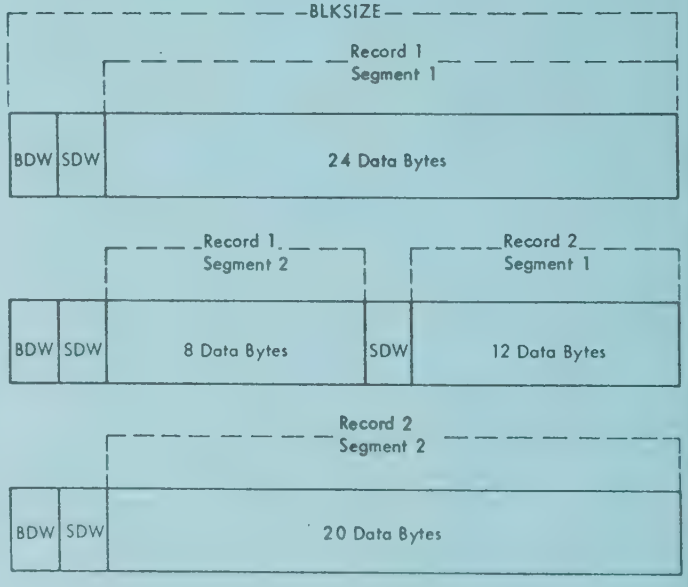


• Figure 40.3. Blocked Records Written Without FORMAT Control

If the logical record is greater than the length of the block, the record is divided into record segments. The number of segments is determined from the READ/WRITE statement and the BLKSIZE specification. The next record, preceded by its SDW, begins in the same block. If the length of the second record exceeds the remainder of the block it too is segmented (see Figure 40.4).

Example: Assume BLKSIZE=32

```
REAL*8 A,B,C,D,E,F,G,H
  .
  .
  .
WRITE(9) A,B,C,D
  .
  .
  .
WRITE(9) E,F,G,H
```



• Figure 40.4. Blocked Segmented Records Written Without FORMAT Control

BACKSPACE Operations

Unblocked Records: For all unblocked records written with or without FORMAT control, the volume is positioned so that the last logical record read or written is transmitted next.

Blocked Records: Blocked records are backspaced on a logical record basis. Thus, a BACKSPACE may result in a deblocking operation rather than making available a new physical record.

Note: Logical records are usually synonymous with the amount of data specified in the I/O list for the READ or WRITE statement that processes the record. Thus, when there is no FORMAT control, the logical record may be spanned over one or more physical records on the volume; however, FORTRAN treats only the logical record as an entity. For records written under

FORMAT control, a single READ/WRITE statement may refer to or create several logical records. This occurs when there is a "/" character in the FORMAT statement or when the I/O list exceeds the FORMAT specifications, causing the FORMAT statement to be used again from the first parenthesis.

Extending a Data Set: The execution of an ENDFILE followed by the execution of a BACKSPACE does not cause the FORTRAN sequence number to be incremented. The data set can be extended (written) using the same FORTRAN sequence number.

Record Length, Buffer Length, and Number of Buffers for Direct Access Data Sets

A direct-access data set can contain only fixed-length, unblocked records. Any attempt to read or write any other record format by specification in the DCB parameter is ignored. The record length and buffer length for a data set are specified by the programmer as the record size in the DEFINE FILE statement, and cannot be changed by specifying the BLKSIZE or LRECL subparameters in the DCB parameter. For example, the statement:

```
DEFINE FILE 8(1000,152,E,INDIC)
```

sets the record length and buffer length permanently at 152 bytes. The direct-access data set defined by this DEFINE FILE statement contains 1000 fixed-length, unblocked records; each record is 152 bytes long and is written under FORMAT control.

The programmer may specify the number of buffers for a direct-access data set as follows:

```
BUFNO=x
```

where: x is the number (1 or 2) of buffers used to read or write the data set.

For records written with FORMAT control, the record format is the same as for fixed-length unblocked records written with FORMAT control for sequential data sets. For records written with no FORMAT control, the records must be fixed-length and unblocked. These records do not contain a block control word or a segment control word.

Spanning Considerations

For records written with no FORMAT control, the input/output list may exceed the logical record length (i.e., block size). In this case a new block is started on output, and the next block is processed on input. If it is shorter than the record length, the remaining portion of the record is padded with zeros (see Figure 41).

The DEFINE FILE field r (r=152 in the example shown above) specifies the maximum size of each record in a data set. It is only when this size is exceeded by the I/O list that spanning occurs.

Note that the spanning feature is an extension to FORTRAN language specifications in that it is applicable only for programs written in FORTRAN under the System/360 Operating System.

When spanning occurs, the FORTRAN library routines do not split an item over the span even if there is enough room in the buffer to accommodate part of the item. The same considerations apply to reading.

Example: A DEFINE FILE statement has specified the record length for a direct-access data set as 20. This statement is then executed:

```
WRITE(9'IX)DP1,DP2,R1,R2
```

where: DP1 and DP2 are real *8 variables.
R1 and R2 are real *4 variables.
IX is an integer variable that contains the record position.

BACKSPACE, END FILE, and REWIND operations are ignored for direct access data sets.

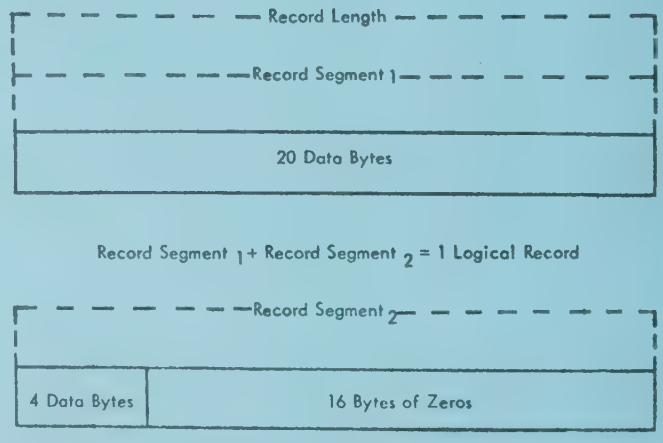


Figure 41. Logical Record (No FORMAT Control) for Direct Access

DCB ASSUMPTIONS FOR LOAD MODULE EXECUTION

For compilation, the LRECL value for the following data sets is fixed and cannot be altered by the programmer:

Data Set	LRECL Value
SYSPRINT	120(G), 137(H)
SYSIN	80
SYSPUNCH	80
SYSLIN	80

The SYSPRINT, SYSIN, and SYSPUNCH compiler data sets can contain blocked records. If the higher level linkage editor (program name: IEWLE440) is used, the SYSLIN data set can contain blocked records.

The BLKSIZE value must be an integral multiple of the corresponding LRECL value shown above. The maximum BLKSIZE value is limited only by the type of input/output device (see Table 16), except that for SYSLIN the maximum BLKSIZE value is 400 with linkage editor IEWLE440.

For load module execution, specifications depend on record type. For F type records, the BLKSIZE value must be an integral multiple of the LRECL value; for V type records, BLKSIZE must be specified as 4 + n x LRECL (where n is the number of records in the block); for U type records, no blocking is permitted. Note, too, that the BLKSIZE and LRECL range is limited only by the type of device used to directly write the data set. Load module DCB parameter default values are shown in Table 17.

Table 16. BLKSIZE Ranges: Device Considerations

Device Type	BLKSIZE Ranges	
	F and U Record Type	V Record Type
Card Reader	1≤x≤80	9≤x≤80
Card Punch	1≤x≤81	9≤x≤89
Printer:		
120 Spaces	1≤x≤121	9≤x≤129
132 Spaces	1≤x≤133	9≤x≤141
144 Spaces	1≤x≤145	9≤x≤153
Magnetic Tape	18≤x≤32,760	
Direct Access:	Without Track Overflow[1]	With Track Overflow[1]
2301	1≤x≤20,483	1≤x≤32,760
2302	1≤x≤4984	1≤x≤32,760
2303	1≤x≤4892	1≤x≤32,760
2311	1≤x≤3625	1≤x≤32,760
2314	1≤x≤7294	1≤x≤32,760

[1]If RECFM=V, the minimum block size is 9.

APPENDICE F

- Dictionnaire des "truffes les plus courantes" -

DICTIONNAIRE DES "TRUFFES" LES PLUS COURANTES

CECI N'EST CERTAINEMENT PAS LA PANACEE POUR TOUTES LES ERREURS, MAIS PLUTOT UNE SUITE DE RECETTES DE CUISINE POUR LES ERREURS LES PLUS FREQUENTES, EN PARTICULIER EN FORTRAN

I. - COMPLETION CODE

CE GENRE DE DIAGNOSTIC APPARAIT DANS LA SYSMSG AVANT LE PROGRAMME FORTRAN, LE PLUS SOUVENT A L'ETAPE GO.

- CC222 CANCEL OPERATEUR :

IL Y A UN MESSAGE SI TROP DE LIGNES OU DE CARTES ONT ETE EDITEES. SINON, VOIR LA PETITE BOITE AU DISPATCHING.

- CC322 TRAPPE DE TEMPS

- VERIFIER QUE LE TEMPS TOTAL CPU EST RAISONNABLE ET/OU QUE L'ON NE FAIT PAS ETERNELLEMENT LA MEME BOUCLE.
- AUGMENTER LE TEMPS DE LA CARTE JOB ET/OU VERIFIER LES BOUCLES.

- CC804 PAS ASSEZ DE PLACE EN MEMOIRE

- CCFCA

- REGARDER A LA FIN DE LA CROSS REFERENCE DU LINK EDIT LA 'TOTAL LENGHT' DU PROGRAMME QUI EST EN HEXADECIMAL :

  ON CONVERTIT EN DECIMAL -----> NOMBRE D'OCTETS
  1 K = 1024 OCTETS -----> NOMBRE DE K

  NE PAS OUBLIER DE RAJOUTER DE LA PLACE POUR LE BUFFER ET LES MODULES D'E/S EN GENERAL # 4K.

- METTRE OU AUGMENTER LE PARAMETRE REGION DE LA CARTE EXEC : REGION.GO = XXXK

- CCX13 X=2,3,... FICHIER MAL CREE OU NON CREE.

et plus particulièrement

CC413 : Ceci arrive lorsque 2 bandes (par ex. la Tape et la FT 14 dans les programmes BSSIO et TEST8 à partir de la Tape) doivent être montées successivement sur le même dérouleur et que la première n'a pas de fin de fichier.

CC613 : le fichier indiqué par LABEL=(n,sl) n'existe pas (la valeur de n est souvent fausse dans ce cas).

CC813 : en lecture, le DSN indiqué dans la carte DD n'est pas le même

CCB37

- Pas assez de place sur un disque magnétique.
- Si ce n'est pas sur un fichier personnel, redonner le job tel quel.

```
*********
* CCCCX *
*   X=1 *
*     4 *
*     5 *
*********
```

UNE INSTRUCTION FORTRAN ENTRAINE UN MAUVAIS POSITIONNEMENT EN MEMOIRE. LE PLUS SOUVENT, CE DIAGNOSTIC EST DU A :

- UNE VARIABLE NON DEFINIE OU NON INITIALISEE.
- UN INDICE DE TABLEAU NON DEFINI OU EGAL A 0.
- UN TABLEAU DEPASSANT LA PLACE RESERVEE DANS L'ORDRE DIMENSION.

II. - DIAGNOSTIC DU LINK EDIT

```
*********
* IEW0132 *
*********
```

IL SE TROUVE APRES LA "CROSS REFERENCE". IL SIGNALE L'UTILISATION D'UNE FONCTION OU D'UN S/P NON DEFINI.

III. - DIAGNOSTICS A L'EXECUTION

ILS APPARAISSENT DANS LE COURANT DES RESULTATS. ILS SONT DE LA FORME IHCXXXI , EN FORTRAN.

```
*********
* IHC2071 *
*********
```

OVERFLOW C'EST A DIRE DEPASSEMENT DE CAPACITE MEMOIRE

UTILISATION D'UN NOMBRE FLOTTANT PLUS GRAND QUE 10**75

- VERIFIER QUE TOUTES LES VARIABLES UTILISEES DANS DES OPERATIONS SONT BIEN INITIALISEES.

EXEMPLE : A=B/C

SI E N'EST PAS DEFINI DANS LE PROGRAMME OU PREND N'IMPORTE QU'ELLE VALEUR EN MEMOIRE, D'OU RESULTATS COMPLETEMENT FARFELUS

- PASSER EN DOUBLE PRECISION POUR VERIFIER QUE L'OVERFLOW N'EST PAS DU A UNE PERTE DE PRECISION EN COURS DE CALCUL.

.B.-

L'EXECUTION CONTINUE EN PRENANT LE PLUS GRAND NOMBRE POSSIBLE QUI PEUT ETRE REPRESENTE.

```
*********
* IHC208I *
*********
```

UNDERFLOW C'EST A DIRE SOUS DEPLACEMENT DE CAPACITE.

UTILISATION D'UN NOMBRE FLOTTANT <10**(-78). MEME PROCESSUS QUE POUR L'OVERFLOW.

.B.-

L'EXECUTION CONTINUE EN REMPLACANT LE NOMBRE EN DEFAUT PAR ZERO.

```
*********
* IHC209I *
*********
```

DIVISION PAR ZERO. L'EXECUTION CONTINUE SANS FAIRE LA DIVISION.

****
- * IHC210I * CE DIAGNOSTIC PEUT AVOIR DIFFERENTES CAUSES
****

. VERIFIER LES ORDRES "EQUIVALENCE"
. DIVISION PAR ZERO POUR UN ENTIER.

****
- * IHC212I * ON CHERCHE A LIRE OU A ECRIRE AVEC UN FORMAT QUI DEPASSE
****
LA LONGUEUR DU BUFFER.

. EN CAS D'ECRITURE OU DE LECTURE SUR DISQUE. VERIFIER LE DEFINE FILE.

. EN CAS DE SORTIE SUR IMPRIMANTE. VERIFIER QUE LE FORMAT NE DEPASSE PAS 132 CARACTERES (OU 120 POUR LES TERMINAUX).

. EN CAS DE BANDE. VERIFIER QU'IL N'Y A PAS DE CONTRADICTION ENTRE LE DCB ET L'ORDRE E/S. CF*

****
- * IHC213I * ON CHERCHE A LIRE OU A ECRIRE SANS UN FORMAT UNE LISTE PLUS
****
GRANDE QUE L'ENREGISTREMENT LOGIQUE. FAIRE LES MEMES VERIFICATIONS QUE POUR IHC212I.

****
- * IHC214I * MAUVAISE LECTURE. ON DEMANDE DE LIRE DU DECIMAL EN FORMAT I.
****
OU ON EST POSITIONNE SUR UN CARACTERE INVALIDE.

. VERIFIER QU'IL NE MANQUE PAS DE CARTES DONNEES OU QU'ELLES NE SONT PAS MELANGEES.

. VERIFIER QU'IL N'Y A PAS UNE MAUVAISE REPRODUCTION DE CARTES (CARACTERE DECALE).

. SUR BANDE. VERIFIER QU'ON EST BIEN POSITIONNE SUR LE BON FICHIER.

****
- * IHC217I * RENCONTRE D'UNE FIN DE FICHIER ALORS QUE LE PROGRAMME PREVOIT
****
DE LIRE DES DONNEES.

. VERIFIER QU'IL NE MANQUE PAS DE CARTES DONNEES.

. SUR BANDE. VERIFIER QUE L'ON NE SE BORNE PAS A LIRE LE LABEL.

****
*CF. FORTRAN 4 G AND H PROGRAMMER'S GUIDE.

Rappel : Le fichier fortran des cartes données est : "FT05"
Le fichier fortran de la liste des résultats est : "FT06"

****
- * IHC219I * EXECUTION TERMINEE. DUE A UN TROP GRAND NOMBRE D'ERREURS
****
(207I OU 208I OU 209I).

A P E N D I C E G

Utilisation de la bibliothéque des programmes "TF"

UTILISATION DE LA BIBLIOTHEQUE DES PROGRAMMES "TF"

Un certain nombre de programmes et sous-programmes ont été mis en bibliothèque sur un disque permanent (c'est-à-dire sans SETUP). Pour avoir accès à cette bibliothèque, il faut mettre devant la carte "EXEC" du programme, la carte JØBLIB décrite ci-dessous :

```
//JØBLIB DD DSN=CIRO719.DELØUIS.TF,DISP=SHR
```

- On peut aussi "concaténer" cette bibliothèque à celle du système (SYSLIB) en faisant précéder la carte //LKED.SYSIN DD * par les deux cartes suivantes :

```
//SYSLIB DD
//       DD DSN=CIRO719.DELOUIS.TF,DISP=SHR        (voir remarque)
```

- On trouvera page 60 la liste des programmes déjà mis en bibliothèque.

I) Cas des programmes directement appelables

Il s'agit des programmes qui se trouvent en bibliothèque sous forme de "load module" et qui sont marqués d'une croix dans la colonne "load" de la liste de la page 60. Ces programmes ont été stockés sur disque avec tous les sous-programmes dont ils ont besoin et ils sont donc directement exécutables. L'étape de "link edit" n'est plus nécessaire, et c'est la carte "JØBLIB" qu'on utilise.

- On trouvera page 62 la liste des cartes contrôle à utiliser pour ces programmes.

II) Cas des autres programmes.

Pour tous les autres programmes, il suffit de concaténer la bibliothèque TF. à celle du système et d'enlever le "deck" correspondant. S'il s'agit d'un programme principal, il faudra utiliser l'instruction INCLUDE placée après la carte //LKED.SYSIN DD *

La liste de la page 60 permet de savoir dans quel cas l'instruction INCLUDE est nécessaire. Les programmes correspondants sont marqués d'une croix en colonne I.

On trouvera page 63 la liste des instructions à utiliser pour les programmes BSS10 et TEST8 les plus courants.

REMARQUE : *Le "deck" si on le laisse, ou si on en met un nouveau, a toujours priorité sur le programme ou sous-programme de même nom se trouvant dans la bibliothèque.*

Si une étape "FORT" de compilation précède le "LINK EDIT" remplacer //SYSLIB DD par //LKED.SYSLIB DD.

LISTE DES PROGRAMMES EN BIBLIOTHEQUE

	LOAD	I (INCLUDE)	COMMENTAIRES
MPLØT			Tracé des FT16
QDRPTE	X		"QUICK" - DERPTE
TEST8SG	X		TEST8S pour "Guelach"
TEST8SV	X		TEST8S " "Verges"
B5CØPIE	X		Copie de plusieurs fichiers d'une "FT12"
C6CØPIE	X		" " " " " "FT16"
DERPTE		X	Programme de pointés de raies
BSS10		X	BSS10 (4K) ne marche qu'avec PLECB2
BSS16K		X	" (16K) " " " "
PP		X	Programme Principal du TEST8 (4K) à partir de la Tape (Sauf Maillard)
PPB5		X	" " " " " " " " FT12
PP16K		X	" " " " (16K) " " " Tape (Sauf Maillard)
PP16KB		X	" " " " " " " " FT12
SP1			Pour TEST8 à partir de la Tape et (PLECB2,FLAG,PLASS)
SP5			SP5 (4K) pour TEST8
SP516K			SP5 (16K) " "
FFT128			Programme de calcul de FFT pour BSS10 et TEST8
PLASS			Sous-programme de lecture des TAPES en assembleur.
FLAG			" " " " "

LISTE DES PROGRAMMES EN BIBLIOTHEQUE

	LOAD	I (INCLUDE)	COMMENTAIRES
DERPTY		X	ANCIEN programme de pointés de raies.
MPLØS	X		Tracé des FT16 de SIG1 à SIG2 de façon exacte (abscisses plus précises)
HERLEC			Programme de lecture des "Tapes" de Mr CONNES (remplace PLECB2)
CPLASS			Sous-programme de lecture des "Tapes" appelé par HERLEC
TEST8SC	X		TEST8S pour Mr CONNES
CPP		X	Programme principal du TEST8 (4 K) à partir de la Tape pour Mr CONNES
CPPB		X	" " " " " FT12 " "
CPP16		X	" " " (16 K) " Tape " "
CPP16B		X	" " " " " FT12 " "
CSP1			Pour TEST8 à partir de la Tape avec HERLEC et CPLASS
CBSS10			BSS10 (4 K) (fait appel à HERLEC)
CBSS16			BSS10 (16 K) " " "
FFT64K			SSP qui calcule la D.F.T. complexe d'une fonction réelle (V3)
ANABRU		X	Programme calculant la val. quodiat. moy. du bruit d'un interf. de Mr CONNES

UTILISATION DES PROGRAMMES DIRECTEMENT CHARGEABLES

TEST 8S (V G C)

```
//DELTEST JOB CIR0719,DELOUIS,TIME=(0,30),MSGLEVEL=(0,0)
//JOBLIB DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
/*SETUP UNIT=BDE7,DDNAME=TAPE,ID=(009234,NORING,SAVE,BLP)
/*SETUP UNIT=BD16,DDNAME=FT12F001,ID=103697
//GO EXEC PGM=TEST8SG,REGION=84K
//GO.TAPE DD   UNIT=BDE7,DISP=(OLD,KEEP),DCB=(DEN=1,TRTCH=ET,BUFNO=1),
// LABEL=(05,NL),DSN=GA
//GO.FT12F001 DD DSN=FT12963,UNIT=BD16,DISP=NEW,LABEL=(05,SL),
// DCB=(RECFM=VS,LRECL=1004,BLKSIZE=1008)
//*GO.FT12F001 DD UNIT=DISK,DISP=(NEW,DELETE),SPACE=(TRK,(80,3)),
//GO.IMP DD SYSOUT=A
//GO.FT05F001 DD DDNAME=SYSIN
//GO.FT06F001 DD SYSOUT=A
//GO.SYSIN DD *
                                                              1100
009630000160000
/*
//
```

MPLOT ou MPLOS

```
//DELMPLO JOB CIR0719,DELOUIS,TIME=(0,58),MSGLEVEL=(0,0)
//JOBLIB DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
/*SETUP UNIT=BD16,DDNAME=FT16F001,ID=104220
/*SETUP UNIT=BD16,DDNAME=FT14F001,ID=102021
//GO EXEC PGM=MPLOT,REGION=78K
//GO.FT14F001 DD UNIT=BDE9,DCB=(RECFM=F,BLKSIZE=4000,DEN=2)
//GO.FT16F001 DD UNIT=BD16,DISP=OLD,LABEL=(1,SL),DSN=FT16001,
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.FT05F001 DD DDNAME=SYSIN
//GO.FT06F001 DD SYSOUT=A
//GO.SYSIN DD *
+0.3507984664000D-05   20    0200001
+0.549300000D+00+0.549800000D+05   08    9    1
+0.10000000D+02+0.168E+01 VERIF TFID4 M=20000
+0.300E+02    4   20
/*
//
```

QUICK DERPTE

```
//DELQDRP JOB CIR0719,DELOUIS,TIME=(0,08),MSGLEVEL=(0,0)
//JOBLIB DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
/*SETUP UNIT=BD16,DDNAME=FT16F001,ID=104220
//GO EXEC PGM=QDRPTE,REGION=114K
//GO.FT16F001 DD UNIT=BD16,DISP=OLD,LABEL=(1,SL),DSN=FT16001,
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.FT05F001 DD DDNAME=SYSIN
//GO.FT06F001 DD SYSOUT=A
//GO.SYSIN DD *
   20+0.3507984664000D-05     2000001
+0.197E+01+0.500E-03
+0.71200000D+03+0.71500000D+03
/*
//
```

B5 COPIE

```
//DELB5CO JOB CIR0719,DELOUIS,TIME=(0,08),MSGLEVEL=(0,0)
/*SETUP UNIT=BD16,DDNAME=FT12F001,ID=(105555,NORING,SAVE,SL)
/*SETUP UNIT=BD16,DDNAME=FT22F001
//JOBLIB DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//GO EXEC PGM=B5COPIE,REGION=72K
//GO.FT12F001 DD UNIT=BD16,DISP=OLD,LABEL=(02,SL),DSN=FT12102,
// DCB=(RECFM=VS,LRECL=1004,BLKSIZE=1008)
//GO.FT12F002 DD UNIT=BD16,DISP=OLD,LABEL=(03,SL),DSN=FT12103,
// DCB=(RECFM=VS,LRECL=1004,BLKSIZE=1008),VOL=REF=*.FT12F001
//*
//GO.FT22F001 DD UNIT=BD16,DISP=NEW,LABEL=(01,SL),DSN=FT12102,
// DCB=(RECFM=VS,LRECL=1004,BLKSIZE=1008)
//GO.FT22F002 DD UNIT=BD16,DISP=NEW,LABEL=(02,SL),DSN=FT12103,
// DCB=(RECFM=VS,LRECL=1004,BLKSIZE=1008),VOL=REF=*.FT22F001
//GO.FT05F001 DD DDNAME=SYSIN
//GO.FT06F001 DD SYSOUT=A
//GO.SYSIN DD *
02
/*
//
```

C6 COPIE

```
//DELC6CO JOB CIR0719,DELOUIS,TIME=(0,08),MSGLEVEL=(0,0)
//JOBLIB DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
/*SETUP UNIT=BD16,DDNAME=FT16F001,ID=(104220,NORING,SAVE,SL)
/*SETUP UNIT=BD16,DDNAME=FT22F001
//GO EXEC PGM=C6COPIE,REGION=60K
//GO.FT16F001 DD UNIT=BD16,DISP=OLD,LABEL=(01,SL),DSN=FT160102,
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.FT16F002 DD UNIT=BD16,DISP=OLD,LABEL=(02,SL),DSN=FT160103,
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032),VOL=REF=*.FT18F001
//*
//GO.FT22F001 DD UNIT=BD16,DISP=NEW,LABEL=(01,SL),DSN=FT160102,
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.FT22F002 DD UNIT=BD16,DISP=NEW,LABEL=(02,SL),DSN=FT160103,
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032),VOL=REF=*.FT22F001
//GO.FT05F001 DD DDNAME=SYSIN
//GO.FT06F001 DD SYSOUT=A
//GO.SYSIN DD *
```

L'APPEL DES PROGRAMMES, NON DIRECTEMENT CHARGEABLES, AU NIVEAU DU "LINK EDIT" (sauf pour Mr CONNES)

TEST8 (4K) A PARTIR DE LA TAPE

```
//SYSLIB DD
//        DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE SYSLIB(PP)
      INSERT PP
      OVERLAY ALPHA
      INSERT SP1,PLECB2,PLASS,FLAG
      OVERLAY ALPHA
      INSERT SP5,FFT128
      ****''DECK DE PLECB2'' ****
/*
```

TEST8 (4K) A PARTIR DE LA FT12

```
//SYSLIB DD
//        DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE SYSLIB(PPB5)
      INSERT PPB5
      OVERLAY ALPHA
      INSERT SP5,FFT128
/*
```

TEST8 (16K) A PARTIR DE LA TAPE

```
//SYSLIB DD
//        DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE SYSLIB(PP16K)
      INSERT PP16K
      OVERLAY ALPHA
      INSERT SP1,PLECB2,PLASS,FLAG
      OVERLAY ALPHA
      INSERT SP516K,FFT128
      ****''DECK DE PLECB2'' ****
/*
```

TEST8 (16K) A PARTIR DE LA FT12

```
//SYSLIB DD
//        DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE SYSLIB(PP16KB)
      INSERT PP16KB
      OVERLAY ALPHA
      INSERT SP516K,FFT128
/*
```

BSS10 (4K) A PARTIR DE LA TAPE

```
//SYSLIB DD
//        DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE SYSLIB(BSS10)
      ****''DECK DE PLECB2'' ****
/*
```

BSS10 (4K) A PARTIR DE LA FT12

```
//SYSLIB DD
//        DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE SYSLIB(BSS10)
      **** ''DECK BIDON'' ****
/*
```

BSS10 (16K) A PARTIR DE LA FT12

```
//SYSLIB DD
//        DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE SYSLIB(BSS16K)
      **** ''DECK BIDON'' ****
/*
```

BSS10 (16K) A PARTIR DE LA TAPE

```
//SYSLIB DD
//        DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE SYSLIB(BSS16K)
      ****''DECK DE PLECB2'' ****
/*
```

```
//DELDUMP JOB CIR0719,DELOUIS,TIME=(0,08),MSGLEVEL=(0,0)
//*          DUMP DE TAPE 9 CNX.
//*          ***********************************
/*SETUP UNIT=BD16,DDNAME=DDTAPE,ID=(009593,NORING,SAVE,BLP)
//JOBLIB DD DSN=BP1.LINKLIB,DISP=SHR
// EXEC PGM=DUMPBAND,PARM='04,HEXA'
//GO.DDTAPE DD UNIT=BD16,LABEL=(01,NL),DISP=(OLD,KEEP),
// DCB=(RECFM=U,BLKSIZE=1672,BUFNO=1,DEN=2)
//GO.DDIMPR DD SYSOUT=A
//GO.SYSUDUMP DD SYSOUT=A
/*
//
```

```
//DELT8S JOB CIR0719,DELOUIS,TIME=(,08),MSGLEVEL=(0,0)
//*          TEST8S
//*          ***********************************
//JOBLIB DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
/*SETUP UNIT=BD16,DDNAME=TAPE,ID=(009593,NORING,SAVE,NL)
//*ETUP UNIT=BD16,DDNAME=FT12F001,ID=108888
//GO EXEC PGM=TEST8SC,REGION=84K
//GO.TAPE DD UNIT=BD16,DISP=(OLD,KEEP),DCB=(DEN=2,BUFNO=1),LABEL=(1,NL)
//GO.FT12F001 DD UNIT=DISK,DISP=(NEW,DELETE),SPACE=(TRK,(99,3)),
//*O.FT12F001 DD UNIT=BD16,DISP=NEW,LABEL=(1,SL),DSN=FT12844,
// DCB=(RECFM=VS,LRECL=1004,BLKSIZE=1008)
//GO.FT05F001 DD DDNAME=SYSIN
//GO.FT06F001 DD SYSOUT=A
//GO.IMP DD SYSOUT=A
//GO.SYSIN DD *
                                                              1100
  152     15000
/*
//
```

```
//DELT1   JOB CIR0719,DELOUIS,TIME=(0,38),MSGLEVEL=(0,0)
//*          TEST8 16K  A PARTIR DE LA TAPE
//*          ***********************************
/*SETUP UNIT=BD16,DDNAME=TAPE,ID=(009593,NORING,SAVE,NL)
/*SETUP UNIT=BD16,DDNAME=FT12F001,ID=105878
/*SETUP  DDNAME=FT14F001,UNIT=BD16,ID=103734
// EXEC FTHLG,PARM.LKED='OVLY,NOLIST,NOXREF',REGION.GO=194K
//SYSLIB DD
//       DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE SYSLIB(CPP16)
      INSERT CPP16
      OVERLAY ALPHA
      INSERT CSP1,HERLEC,CPLASS
      OVERLAY ALPHA
      INSERT SP516K,FFT128
/*
//GO.TAPE DD UNIT=BD16,DISP=(OLD,KEEP),DCB=(DEN=2,BUFNO=1),LABEL=(1,NL)
//GO.FT12F001 DD UNIT=BD16,DISP=(NEW,KEEP),LABEL=(01,SL),DSN=FT12022,
// DCB=(BUFNO=2,RECFM=VS,LRECL=1004,BLKSIZE=1008)
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2)
//GO.IMP DD SYSOUT=A
//GO.SYSIN DD *
                                                             11000
   22    120000    1+0.11530000D+04
+0.70000000D+01+0.80000000D+05   16    1    1
+0.10000000D+02+0.250E+02 BSSE10   22
/*
//
```

```
//DELT2   JOB CIR0719,DELOUIS,TIME=(0,28),MSGLEVEL=(0,0)
//*          TEST8 16K  A PARTIR DE LA FT12
//*          ***********************************
/*SETUP UNIT=BD16,DDNAME=FT12F001,ID=105878
/*SETUP  DDNAME=FT14F001,UNIT=BD16,ID=103734
// EXEC FTHLG,PARM.LKED='OVLY,NOLIST,NOXREF',REGION.GO=192K
//SYSLIB DD
//       DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE SYSLIB(CPP16B)
      INSERT CPP16B
      OVERLAY ALPHA
      INSERT SP516K,FFT128
/*
//GO.FT12F001 DD UNIT=BD16,DISP=(NEW,KEEP),LABEL=(01,SL),DSN=FT12022,
// DCB=(BUFNO=2,RECFM=VS,LRECL=1004,BLKSIZE=1008)
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2)
//GO.IMP DD SYSOUT=A
//GO.SYSIN DD *
                                                             01000
   22    110000    1+0.11530000D+04
+0.70000000D+01+0.80000000D+05   16    1    1
```

PROGRAMMES POUR M. CONNES

```
//DELT3   JOB CIR0719,DELOUIS,TIME=(0,18),MSGLEVEL=(0,0)
//*         BSS10  16K A PARTIR DE LA TAPE
//*         ***********************************
/*SETUP UNIT=BD16,DDNAME=TAPE,ID=(009593,NORING,SAVE,NL)
/*SETUP  DDNAME=FT14F001,UNIT=BD16,ID=103734
// EXEC FTHLG,PARM.LKED='NOLIST,NOXREF',REGION.GO=162K
//SYSLIB DD
//       DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE SYSLIB(CBSS16)
/*
//GO.TAPE DD UNIT=BD16,DISP=(OLD,KEEP),DCB=(DEN=2,BUFNO=1),LABEL=(1,NL)
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2)
//GO.IMP DD SYSOUT=A
//GO.SYSIN DD *
                                                          10100
+0.50000000D+04      9000    1   22
+0.70000000D+01+0.80000000D+05   16    1    1    0
+0.10000000D+03+0.250E+02 BSSE10   22
/*
//
```

```
//DELT4   JOB CIR0719,DELOUIS,TIME=(0,18),MSGLEVEL=(0,0)
//*         BSS10  16K A PARTIR DE LA FT12
//*         ***********************************
/*SETUP UNIT=BD16,DDNAME=FT12F001,ID=105878
/*SETUP  DDNAME=FT14F001,UNIT=BD16,ID=103734
// EXEC FTHLG,PARM.LKED='NOLIST,NOXREF',REGION.GO=156K
//SYSLIB DD
//       DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE SYSLIB(CBSS16)
BESD       -   AHERLEC          CPLASS                                  0001
BEND                                F15OCT70 14.13  6/01/72             0002
/*
//GO.FT12F001 DD UNIT=BD16,DISP=(NEW,KEEP),LABEL=(01,SL),DSN=FT12022,
// DCB=(BUFNO=2,RECFM=VS,LRECL=1004,BLKSIZE=1008)
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2)
//GO.IMP DD SYSOUT=A
//GO.SYSIN DD *
                                                          00100
+0.50000000D+04      9000    1   22
+0.70000000D+01+0.80000000D+05   16    1    1    0
+0.10000000D+03+0.250E+02 BSSE10   22
/*
//
```

```
//DELT5   JOB CIR0719,DELOUIS,TIME=(0,18),MSGLEVEL=(0,0)
//*           TEST8  4K  A PARTIR DE LA TAPE
//*           ***********************************
/*SETUP UNIT=BD16,DDNAME=TAPE,ID=(009593,NORING,SAVE,NL)
/*SETUP UNIT=BD16,DDNAME=FT12F001,ID=105878
/*SETUP  DDNAME=FT14F001,UNIT=BD16,ID=103734
// EXEC FTHLG,PARM.LKED='OVLY,NOLIST,NOXREF',REGION.GO=132K
//SYSLIB DD
//       DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE SYSLIB(CPP)
      INSERT CPP
      OVERLAY ALPHA
      INSERT CSP1,HERLEC,CPLASS
      OVERLAY ALPHA
      INSERT SP5,FFT128
/*
//GO.TAPE DD UNIT=BD16,DISP=(OLD,KEEP),DCB=(DEN=2,BUFNO=1),LABEL=(1,NL)
//GO.FT12F001 DD UNIT=BD16,DISP=(NEW,KEEP),LABEL=(02,SL),DSN=FT12023,
// DCB=(BUFNO=2,RECFM=VS,LRECL=1004,BLKSIZE=1008)
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2)
//GO.IMP DD SYSOUT=A
//GO.SYSIN DD *
                                                          11000
   22    120000    1+0.11530000D+04
+0.70000000D+01+0.80000000D+05   16    1    1
+0.10000000D+02+0.250E+02 BSSE10   22
/*
//
```

PROGRAMMES POUR M. CONNES

```
//DELT6   JOB CIR0719,DELOUIS,TIME=(0,18),MSGLEVEL=(0,0)
//*          TEST8  4K  A PARTIR DE LA FT12
//*          **********************************
/*SETUP UNIT=BD16,DDNAME=FT12F001,ID=105878
/*SETUP  DDNAME=FT14F001,UNIT=BD16,ID=103734
// EXEC FTHLG,PARM.LKED='OVLY,NOLIST,NOXREF',REGION.GO=132K
//SYSLIB DD
//       DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE SYSLIB(CPPB)
      INSERT CPPB
      OVERLAY ALPHA
      INSERT SP5,FFT128
/*
//GO.FT12F001 DD UNIT=BD16,DISP=(NEW,KEEP),LABEL=(01,SL),DSN=FT12022,
// DCB=(BUFNO=2,RECFM=VS,LRECL=1004,BLKSIZE=1008)
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2)
//GO.IMP DD SYSOUT=A
//GO.SYSIN DD *
                                                              01000
   22    120000    1+0.11530000D+04
+0.70000000D+01+0.80000000D+05   16    1    1
+0.10000000D+02+0.250E+02 BSSE10   22
/*
//

//DELT7   JOB CIR0719,DELOUIS,TIME=(0,08),MSGLEVEL=(0,0)
//*          BSS10   4K A PARTIR DE LA TAPE
//*          **********************************
/*SETUP UNIT=BD16,DDNAME=TAPE,ID=(009593,NORING,SAVE,NL)
/*SETUP  DDNAME=FT14F001,UNIT=BD16,ID=103734
// EXEC FTHLG,PARM.LKED='NOLIST,NOXREF',REGION.GO=114K
//SYSLIB DD
//       DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE SYSLIB(CBSS10)
/*
//GO.TAPE DD UNIT=BD16,DISP=(OLD,KEEP),DCB=(DEN=2,BUFNO=1),LABEL=(1,NL)
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2)
//GO.IMP DD SYSOUT=A
//GO.SYSIN DD *
                                                              10100
+0.50000000D+04      4000    1   22
+0.70000000D+01+0.80000000D+05   16    1    1    0
+0.10000000D+03+0.250E+02 BSSE10   22
/*
//

//DELT8   JOB CIR0719,DELOUIS,TIME=(0,08),MSGLEVEL=(0,0)
//*          BSS10   4K A PARTIR DE LA FT12
//*          **********************************
/*SETUP UNIT=BD16,DDNAME=FT12F001,ID=105878
/*SETUP  DDNAME=FT14F001,UNIT=BD16,ID=103734
// EXEC FTHLG,PARM.LKED='NOLIST,NOXREF',REGION.GO=110K
//SYSLIB DD
//       DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE SYSLIB(CBSS10)
BESD        -    AHERLEC          CPLASS                          0001
BEND                                  F150CT70 14.13  6/01/72     0002
/*
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2)
//GO.FT12F001 DD UNIT=BD16,DISP=(OLD,KEEP),LABEL=(01,SL),DSN=FT12022,
// DCB=(BUFNO=2,RECFM=VS,LRECL=1004,BLKSIZE=1008)
//GO.IMP DD SYSOUT=A
//GO.SYSIN DD *
                                                              00100
+0.50000000D+04      3910    1   22
+0.70000000D+01+0.80000000D+05   16    1    1    0
+0.10000000D+03+0.250E+02 FIN  FIN  FIN  FIN  FIN
/*
```

A P P E N D I C E H

Exécutions successives du même programme en un seul job

EXECUTIONS SUCCESSIVES DU MEME PROGRAMME EN UN SEUL JOB

CAS D'UN PROGRAMME DIRECT. CHARGEABLE. (MPLOS AVEC 3 CALCULS)

```
//MPLOS JOB SPM4444,DUPONT,TIME=(1,30),MSGLEVEL=(0,0)
//*      MPLOS
//*      *******
//JOBLIB DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
/*SETUP UNIT=BD16,DDNAME=FT16F001,ID=104220
/*SETUP UNIT=BD16,DDNAME=FT14F001,ID=105269
//GO EXEC PGM=MPLOS,REGION=86K
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2),LABEL=(1)
//GO.FT16F001 DD UNIT=BD16,DISP=OLD,LABEL=(1,SL),DSN=FT160102,
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.FT05F001 DD DDNAME=SYSIN
//GO.FT06F001 DD SYSOUT=A
//GO.SYSIN DD *
+0.3507984600000D-05 1280     0200001
+0.16524400D+05+0.16525200D+05     1    1    1
+0.50000000D-02+0.180E+01 2
+0.560E+02     1   20
/*
//*
//GO EXEC PGM=MPLOS,REGION=86K
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2),LABEL=(2)
//GO.FT16F001 DD UNIT=BD16,DISP=OLD,LABEL=(2,SL),DSN=FT160103,
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.FT05F001 DD DDNAME=SYSIN
//GO.FT06F001 DD SYSOUT=A
//GO.SYSIN DD *
+0.3507984600000D-05 1280     0200001
+0.16524400D+05+0.16525200D+05     1    1    2
+0.50000000D-02+0.182E+01  3
+0.560E+02     1   20
/*
//*
//GO EXEC PGM=MPLOS,REGION=86K
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2),LABEL=(3)
//GO.FT16F001 DD UNIT=BD16,DISP=OLD,LABEL=(3,SL),DSN=FT160104,
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.FT05F001 DD DDNAME=SYSIN
//GO.FT06F001 DD SYSOUT=A
//GO.SYSIN DD *
+0.3507984600000D-05 1280     0200001
+0.16524400D+05+0.16525200D+05     1    1    3
+0.50000000D-02+0.183E+01    4
+0.560E+02     1   20
/*
//
```

CAS D'UN PROGRAMME APPELE PAR UN ''INCLUDE''. (DERPTE AVEC 3 CALCULS)

```
//DERPTE JOB CIR0999,DELOUIS,TIME=(0,40),MSGLEVEL=(0,0)
//*        DERPTE
//*      **********
/*FORMAT PR,DDNAME=FT06F001,OVFL=ON,COPIES=1
/*SETUP UNIT=BD16,DDNAME=FT16F001,ID=104220
//E EXEC FTHLG,PARM.LKED='NOLIST,NOXREF',REGION.GO=146K
//SYSLIB DD
//       DD DSN=CIR0719.DELOUIS.TF,DISP=SHR
//LKED.SYSIN DD *
     INCLUDE SYSLIB(DERPTE)
/*
//GO.FT16F001 DD UNIT=BD16,DISP=OLD,LABEL=(1,SL),DSN=FT160102,
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.SYSIN DD *
 1280+0.3507984600000D-05      20000000100
+0.180E+01+0.900E-01+0.200E+01
+0.16524000D+05+0.16526000D+05
+0.00000000D+00+0.00000000D+00 ((COEF CC1 ET CC2 (SIGC=SIG*(1+CC1)+CC2  ))
+0.10000000D+00+0.280E+02     1
/*
//*
// EXEC PGM=*.E.LKED.SYSLMOD,REGION=146K
//GO.FT05F001 DD DDNAME=SYSIN
//GO.FT06F001 DD SYSOUT=A
//GO.FT16F001 DD UNIT=BD16,DISP=OLD,LABEL=(2,SL),DSN=FT160103,
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.SYSIN DD *
 1280+0.3507984600000D-05      20000000100
+0.180E+01+0.900E-01+0.200E+01
+0.16524000D+05+0.16526000D+05
+0.00000000D+00+0.00000000D+00 ((COEF CC1 ET CC2 (SIGC=SIG*(1+CC1)+CC2  ))
+0.10000000D+00+0.280E+02     1
/*
//*
// EXEC PGM=*.E.LKED.SYSLMOD,REGION=146K
//GO.FT05F001 DD DDNAME=SYSIN
//GO.FT06F001 DD SYSOUT=A
//GO.FT16F001 DD UNIT=BD16,DISP=OLD,LABEL=(3,SL),DSN=FT160104,
// DCB=(RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.SYSIN DD *
 1280+0.3507984600000D-05      20000000100
+0.180E+01+0.900E-01+0.200E+01
+0.16524000D+05+0.16526000D+05
+0.00000000D+00+0.00000000D+00 ((COEF CC1 ET CC2 (SIGC=SIG*(1+CC1)+CC2  ))
```

CAS PARTICULIER DU PROGRAMME TFID-T102 (3 CALCULS SUCCESSIFS)

```
//TFIDT102 JOB COT0830,MAILLARD,TIME=(1,30),MSGLEVEL=(0,0)
//*      TFID-T102
//*
/*SETUP DDNAME=FT01F001,UNIT=3330,ID=SET001
/*SETUP UNIT=BD16,DDNAME=FT32F001,ID=105555
/*SETUP UNIT=BD16,DDNAME=FT16F001,ID=104220
/*SETUP  DDNAME=FT14F001,UNIT=BD16,ID=102021
//S1 EXEC FTHLG,PARM.LKED='NOXREF,NOLIST',REGION.GO=194K
//LKED.SYSLIN DD DSN=CIR0719.DELOUIS.TFID4,UNIT=3330,VOL=SER=SET001,
// DISP=OLD
//LKED.SYSLMOD DD DISP=(NEW,PASS),DSN=+GOSET1(MAIN)
//GO.FT01F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF01Y,VOL=SER=SET001
//GO.FT11F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF11Y,VOL=SER=SET001
//GO.FT21F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF21Y,VOL=SER=SET001
//GO.FT22F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(57,1)),
// DSNAME=CIR0719.DELOUIS.TF22Y,VOL=SER=SET001
//GO.FT32F001 DD UNIT=BD16,DISP=OLD,LABEL=(02,SL),DSN=FT12102,
// DCB=(BUFNO=2,RECFM=VS,LRECL=1004,BLKSIZE=1008)
//GO.SYSIN DD *
    1     20000
/*
//S2 EXEC FTHLG,PARM.LKED='NOXREF,NOLIST',REGION.GO=172K
//LKED.SYSLIN DD DSN=CIR0719.DELOUIS.T102,UNIT=3330,VOL=SER=SET001,
// DISP=OLD
//LKED.SYSLMOD DD DISP=(NEW,PASS),DSN=+GOSET2(MAIN)
//GO.FT01F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF01Y,VOL=SER=SET001
//GO.FT11F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF11Y,VOL=SER=SET001
//GO.FT16F001 DD UNIT=BD16,DISP=NEW,LABEL=(1,SL),DSN=FT160102,
// DCB=(BUFNO=2,RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2),LABEL=(1)
//GO.SYSIN DD *
     20000    32  102     1     1     0
 1280+0.16550000D+051
/*
//*
// EXEC PGM=*.S1.LKED.SYSLMOD,REGION=194K
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//GO.FT01F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF01Y,VOL=SER=SET001
//GO.FT11F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF11Y,VOL=SER=SET001
//GO.FT21F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF21Y,VOL=SER=SET001
//GO.FT22F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(57,1)),
// DSNAME=CIR0719.DELOUIS.TF22Y,VOL=SER=SET001
//GO.FT32F001 DD UNIT=BD16,DISP=OLD,LABEL=(03,SL),DSN=FT12103,
// VOL=SER=103729,                                              *
// DCB=(BUFNO=2,RECFM=VS,LRECL=1004,BLKSIZE=1008)
//GO.SYSIN DD *
    1     20000
/*
// EXEC PGM=*.S2.LKED.SYSLMOD,REGION=194K
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//GO.FT01F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF01Y,VOL=SER=SET001
//GO.FT11F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF11Y,VOL=SER=SET001
//GO.FT16F001 DD UNIT=BD16,DISP=NEW,LABEL=(2,SL),DSN=FT160103,
// DCB=(BUFNO=2,RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2),LABEL=(2)
//GO.SYSIN DD *
     20000    32  103     1     2     0
 1280+0.16550000D+051
/*
//*
// EXEC PGM=*.S1.LKED.SYSLMOD,REGION=194K
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//GO.FT01F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF01Y,VOL=SER=SET001
//GO.FT11F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF11Y,VOL=SER=SET001
//GO.FT21F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF21Y,VOL=SER=SET001
//GO.FT22F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(57,1)),
// DSNAME=CIR0719.DELOUIS.TF22Y,VOL=SER=SET001
//GO.FT32F001 DD UNIT=BD16,DISP=OLD,LABEL=(04,SL),DSN=FT12104,
// DCB=(BUFNO=2,RECFM=VS,LRECL=1004,BLKSIZE=1008)
//GO.SYSIN DD *
    1     20000
/*
// EXEC PGM=*.S2.LKED.SYSLMOD,REGION=194K
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//GO.FT01F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF01Y,VOL=SER=SET001
//GO.FT11F001 DD UNIT=3330,DISP=(OLD,KEEP),SPACE=(TRK,(1024,1)),
// DSNAME=CIR0719.DELOUIS.TF11Y,VOL=SER=SET001
//GO.FT16F001 DD UNIT=BD16,DISP=NEW,LABEL=(3,SL),DSN=FT160104,
// DCB=(BUFNO=2,RECFM=VS,LRECL=1028,BLKSIZE=1032)
//GO.FT14F001 DD UNIT=BD16,DCB=(RECFM=F,BLKSIZE=4000,DEN=2),LABEL=(3)
//GO.SYSIN DD *
     20000    32  104     1     3     0
 1280+0.16550000D+051
/*
//
```

A P P E N D I C E I

Particularités des programmes adaptés pour
l'exploitation des interférogrammes enregistrés en 9 pistes
avec le nouvel interféromètre de Mr CONNES.

- Valeur de FRANGE = + 0.79123932 D-05

- "TAPE" il s'agit d'une bande 9 pistes, 800 bpi, NL. (BD16,DEN=2)

  Les blocs sont de 170 mesures et chaque mesure est codée sur 3 octets. Le détail du codage sera donné plus loin p. 70.

- Remplacer "T102" par "T103".

- Il faut compter environ 3 secondes pour décoder et recopier 100 000 valeurs de la "TAPE"sur 370/165.

- IMP = 0 seules les 10 premières valeurs de chaque bloc de 10 000 valeurs sont listées.

- IMP = 1 liste des blocs décodés par (Herlec + Plass).

- IMP = 2 liste et dump décodés par (Herlec + Plass).

- Pour tester les programmes, il existe une "TAPE" spéciale (n° 9593) où il y a un interf. "synthétique" de 10^6 valeurs. (Voir détails sur les feuilles de classification des bandes).

Exemples de durée sur 370/165 I.B.M. :

- BSS10 - 16 K (TAPE) M = 8 000 IMP = 0

  5 sec. (160 K) 118 lignes

- TEST8S - M = 10^6

  30 sec.

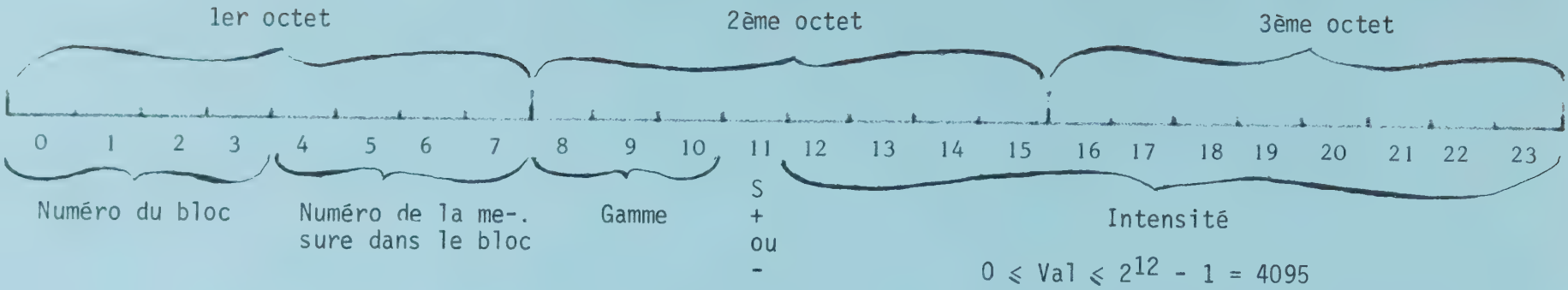
- BSS10 - 4 K - M = 4 000 (FT12)

  3 sec.

- TEST8 - 16 K (FT12) M = 10^6

  20 sec.

- TFID3 - T103 → 2'30" pour M = 500 000



- Les blocs sont numérotés cycliquement : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1 etc.....

- Les mesures sont numérotées cycliquement : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1 etc.....

- Signe { bit 11 égal à 1 → négatif
  bit 11 égal à 0 → positif

- Mesure le bit 12 est le bit de plus fort poids.

- Gamme :

Valeur de la gamme	Gain enregistrement	Gain lecture	Bits 8 9 10
1	1	1024	0 0 1
2	4	256	0 1 0
3	16	64	0 1 1
4	64	16	1 0 0
5	256	4	1 0 1
6	1024	1	1 1 0

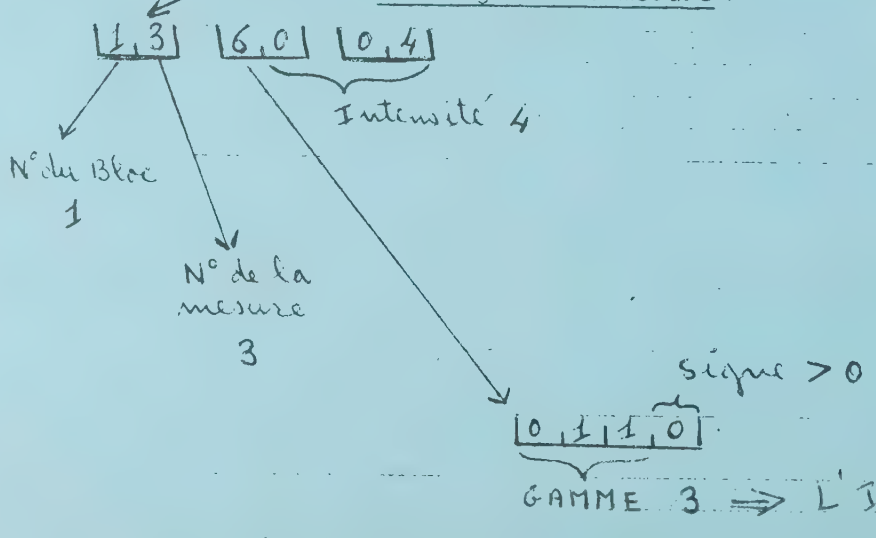
DUMP D'UN BLOC

```
!            2            3            4            5            6
I************I************I************I************I************I************I
```

```
106003116004 126003136003 146004156003 166003176004 186003196004 106003116003
126004136003 146003156003 166004176004 186003196003 106003116004 126003136004
146003156004 166003176003 186003196004 106004116003 126003136003 146004156003
166003176004 186003196004 106003116003 126004136003 146004156003 166003176004
186003196004 106003116004 126004136003 146003156004 166003176003 186004196003
106004116003 126003136003 146004156004 166003176003 186004196003 106003116004
126003136004 146003156003 166003176003 186004196004 106002116004 126003136003
146004156003 166003176004 186003196003 106003116004 126003136003 146003156004
166003176003 186004196003 106004116003 126003136004 146003156003 166004176003
186004196003 106003116003 126004136003 146004156003 166004176003 186003196003
106004116003 126004136003 146003156004 166003176004 186003196003 106004116003
126003136003 146004156003 166004176003 186002196003 106004116003 126004136004
146003156003 166003176004 186003196004 106003116003 126003136003 146004156003
166004176003 186003196004 106003116003 126004136003 146003156004 166003176004
186003196004
```

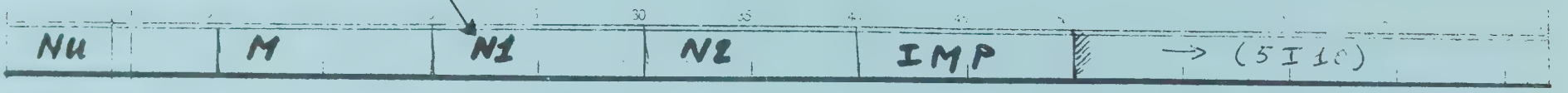
```
++++++++++++++++++++++++++++++++++++++++++++++++++TAPE MARK++++++++++++++++++++++++++++++++++++++++++++++++++
```

Décodage d'une mesure



"ANABRUINT"

carte donnée




```
C
C    *****  PROGRAMME  ANABRUINT  *****
C       IL CALCULE LA VAL. QUADRATIQUE MOYENNE DU BRUIT (VQM) D'UN INTERFEROGRAMME
C    ENTRE LES BLOCS DE 17? VALEURS N1 ET N2 COMPRIS. AUPARAVANT, IL CALCULE LA VA-
C    LEUR MOYENNE DM ET LA SOUSTRAIT.
C *** SI IMP=0  PAS D'AUTRE LISTE.
C *** SI IMP=1  LISTE DE CHAQUE BLOC DE NS VAL. DE LA FT13 LU PAR LE SSP HERSEC.
C *** SI IMP=2  DUMP ET LISTE ''      '''     ''     ''     ''     ''     ''
C
```

→ enregistré en 9 pistes

. Si en lisant la bande, le programme rencontre une fin de fichier (E.O.F.) avant d'arriver au bloc N1, il arrête le calcul en sortant un message.

. Si une "E.O.F." est trouvée entre les blocs N1 et N2, le même message est sorti, mais le calcul de la valeur quadratique moyenne (V.Q.M.) est fait sur les "NVAL" valeurs déjà lués.

. Le calcul de V.Q.M. est fait sur un maximum de 10.000 valeurs de l'interférogramme. La valeur de N2 est diminuée automatiquement si cette contrainte n'est pas respectée ou si N2 est incompatible avec M, le nombre de valeurs de l'interf., qui est lu dans la carte donnée.

. La valeur moyenne est sortie en même temps que V.Q.M.

APPENDICE J

Le paramètre "IAP" et les différentes apodisations possibles

La pondération de l'interférogramme I (δ) est choisie en fonction de la valeur du paramètre "IAP" des cartes données. Les valeurs primaires du spectre qui sont stockées sur bande magnétique (FT16) correspondent toujours à l'interférogramme non pondéré ce qui laisse le choix de l'apodisation au moment de l'exploitation du spectre. En effet, l'apodisation est toujours réalisée au niveau de l'interpollation du spectre en le convoluant par la T.F. de la fonction P (δ) qui aurait permisde pondérer directement l'interférogramme.

Soit δM la différence de marche maximum ($\delta \leq \delta M$)

- Si IAP = 0 : (S.A.) Pas d'apodisation et P (δ) = 1

- Si IAP = 1 : (APØ1) et $P(\delta) = (1 - \frac{\delta^2}{\delta M^2})^2 = F^2$

  en posant $F = 1 - \frac{\delta^2}{\delta M^2}$

- Si IAP = 2 : (APØ2)

  $P(\delta) = 0{,}0627 + 0{,}3321 \times F + 0{,}2184 \times F^2 + 0{,}3868\ F^3$

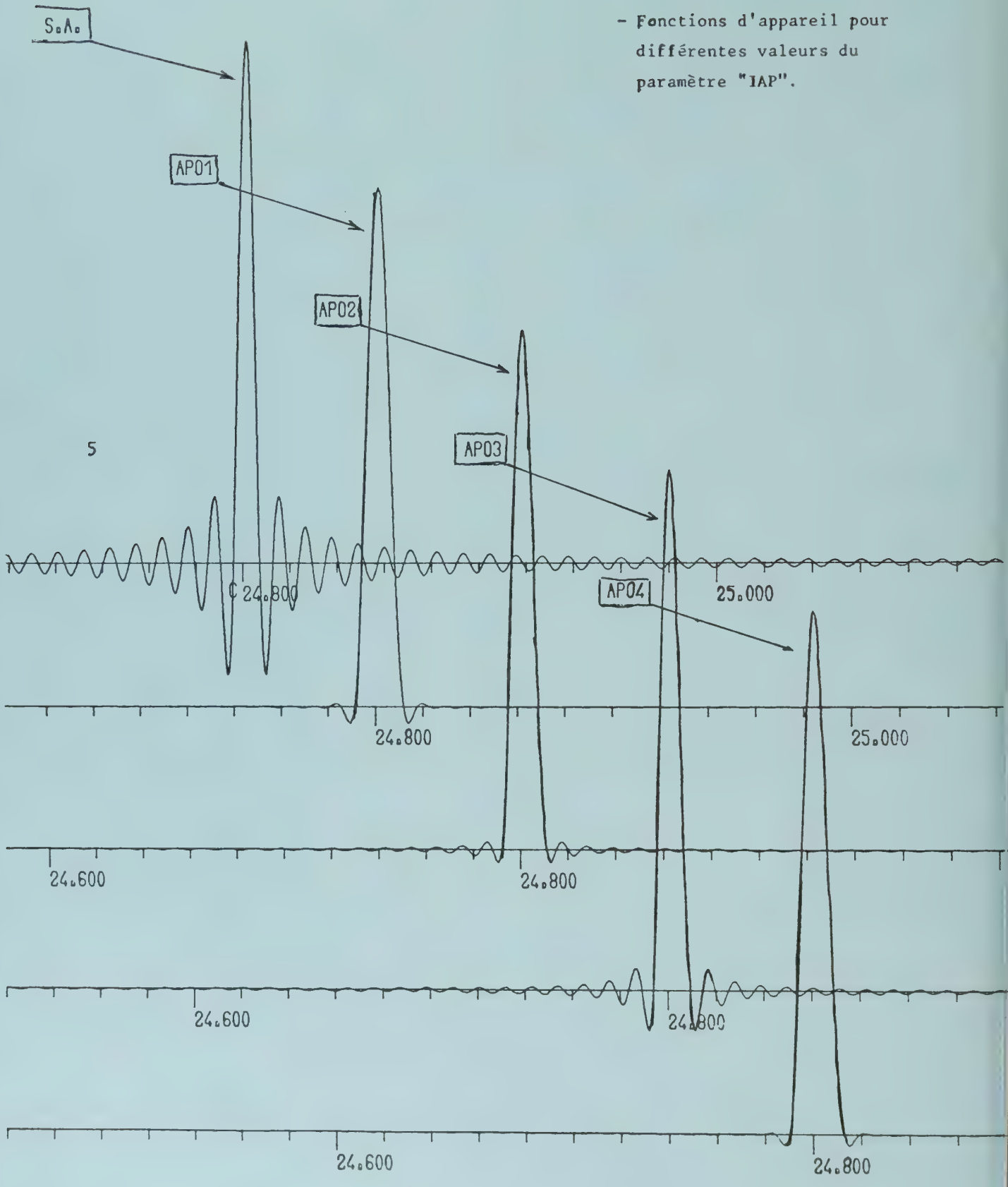
- Si IAP = 3 : (APØ3)

  $P(\delta) = 0{,}2398 + 0{,}4581 \times F + 0{,}2250 \times F^2 + 0{,}0772 \times F^3$

- Si IAP = 4 : (APØ4), $P(\delta) = F^{5/2}$

  Les figures de la page suivante, donnent l'allure de la fonction d'appareil pour ces différentes apodisations.

REMARQUE : Les apodisations du type "APØ2", "APØ3", "APØ4" ne sont disponibles que dans les programmes : "T102, T103, MPLØT, MPLØS, DERPTE, QDRPTE".



- Fonctions d'appareil pour différentes valeurs du paramètre "IAP".

Service de Reprographie
C . N . R . S .
Gif